



TUGAS AKHIR-TE 141599

***WAYPOINT TRACKING CONTROL PADA QUADCOPTER
MENGUNAKAN PID-BASED SLIDING MODE CONTROL***

Yurid Eka Nugraha
NRP 2212 100 119

Dosen Pembimbing
Ir. Rusdhianto Effendie AK., MT.
Eka Iskandar, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT -TE 141599

***WAYPOINT TRACKING CONTROL OF QUADCOPTER USING
PID-BASED SLIDING MODE CONTROL***

Yurid Eka Nugraha
NRP 2212 100 119

Advisor
Ir. Rusdhianto Effendie AK., MT.
Eka Iskandar, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh November Institute of Technology
Surabaya 2016

**WAYPOINT TRACKING CONTROL PADA QUADCOPTER
MENGUNAKAN PID-BASED SLIDING MODE CONTROL**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Rusdhianto Effendie, AK, MT

NIP. 195704241985021001

Eka Iskandar, ST, MT,

NIP. 1980 05282008 121 001



JANAURI, 2016

WAYPOINT TRACKING CONTROL PADA QUADCOPTER MENGUNAKAN PID-BASED SLIDING MODE CONTROL

Nama : Yurid Eka Nugraha
Pembimbing : Ir. Rusdhianto Effendie A.K., MT.
Eka Iskandar, ST., MT.

ABSTRAK

Quadcopter merupakan suatu sistem yang bersifat nonlinear dan tidak stabil. Oleh karena itu, dibutuhkan sebuah kontroler yang mampu menjaga kondisi plant tetap stabil dan mengikuti referensi yang diberikan. *Sliding Mode Control* dapat mengatasi perubahan parameter *plant* tanpa menyebabkan gangguan pada performa sistem. *Sliding Mode Control* merupakan metode kontrol berbasis *switching* berfrekuensi tinggi dan memiliki aksi kontrol yang sifatnya diskontinyu. Namun, *Sliding Mode Control* dapat menimbulkan *chattering* pada sistem. Oleh karena itu, pada penelitian ini digunakan metode *Sliding Mode Control* dengan sinyal kontrol natural berupa struktur kontroler PID pada *plant quadcopter*. Dengan menggunakan metode ini, diharapkan *quadcopter* dapat melakukan *waypoint tracking* dengan akurat dari referensi yang diharapkan tanpa adanya *chattering*. Kestabilan dari keseluruhan *loop* pengaturan sistem dipastikan menggunakan fungsi kestabilan Lyapunov. Kontroler hasil desain diaplikasikan pada sistem *quadcopter*. Hasil simulasi menunjukkan bahwa sistem *quadcopter* dengan menggunakan kontroler PID-Based *Sliding Mode Control* mampu melakukan *tracking* terhadap sumbu x dan y dengan *error RMS* masing-masing 0,2% dan 0,47%.

Kata Kunci : PID, *Quadcopter*, *Sliding Mode Control*, *Tracking*

WAYPOINT TRACKING CONTROL OF QUADCOPTER USING PID-BASED SLIDING MODE CONTROL

Name : Yurid Eka Nugraha
Advisor : Ir. Rusdhianto Effendie AK., MT.
Eka Iskandar, ST., MT.

ABSTRACT

Quadcopter is a system that is both nonlinear and unstable. Therefore, it needs a controller that is capable of maintaining the plant's condition remains stable and follow the references given. Sliding Mode Control can overcome changes of parameter of the plant without causing disruption on the performance of the system. Sliding Mode Control is a method which using high-frequency switching and have discontinuous control action.. However, Sliding Mode Control can cause chattering on the system. Therefore, in this study it is used Sliding Mode Control with natural control signals in the form of the structure of PID controller on quadcopter. By using this method, it is expected that quadcopter can do waypoint reference tracking without having chattering phenomenon. The stability of the entire control system loop is ensured using Lyapunov stability functions. Results of controller design is applied to quadcopter system. Simulation results indicate that the system is capable of reference tracking in quadcopter system in x and y axis with RMS error each 0.2% and 0.47%..

Keywords: *PID Controller, Quadcopter, Sliding Mode Control, Tracking*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan anugerah-Nya.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan S-1 pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya .

Penulis menyadari bahwa dalam penyelesaian pengerjaan Tugas Akhir ini tidak terlepas dari bantuan dari berbagai pihak. Untuk itu dengan segala hormat, penulis mengucapkan terima kasih kepada semua pihak yang telah berperan dalam penyelesaian Tugas Akhir ini,

Penulis memohon maaf atas segala kekurangan pada Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi pengembangan keilmuan di kemudian hari.

Surabaya, Januari 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
HALAMAN PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Permasalahan	2
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Metodologi.....	4
1.6. Sistematika.....	5
1.7. Relevansi.....	6
BAB II TEORI DASAR.....	7
2.1. Tinjauan Pustaka	7
2.1.1. Konsep Dasar	8
2.1.2. Kinematika <i>Quadcopter</i>	10
2.1.3. Dinamika <i>Quadcopter</i>	16
2.1.4. Model Matematika	24
2.2. Spesifikasi <i>Plant Yang Digunakan</i>	25
2.2.1. Sensor <i>Gyro</i> dan <i>Accelerometer</i>	25
2.2.2. <i>Ardupilot Mega</i>	26
2.2.3. <i>Transmitter</i> dan <i>Receiver Radio</i>	27
2.2.4. <i>Electronic Speed Controller (ESC)</i>	28
2.2.5. Motor BLDC dan <i>Propeller</i>	29
2.2.6. <i>Ground Station</i>	29
2.3. <i>PID Controller</i>	30
2.3.1. Proporsional.....	30
2.3.2. Integral.....	32
2.3.3. Derivatif.....	32
2.4. <i>Sliding Mode Control</i>	33
2.4.1. Pengenalan <i>Sliding Mode Control</i>	34

2.4.2. Analisis Kestabilan Lyapunov	35
2.4.3. Permukaan Luncur	37
2.4.4. Sinyal Kontrol	39
2.4.5. <i>Sliding Mode Control</i> Berbasis PID	39
2.5. Linearisasi	40
BAB III PERANCANGAN SISTEM.....	41
3.1. Spesifikasi Sistem	41
3.2. Identifikasi Kebutuhan	41
3.3. Desain Mekanik dan Elektronik Sistem	41
3.3.1. Desain Mekanik	42
3.3.2. Desain Elektronik	42
3.4. Pemodelan Motor- <i>Propeller</i>	42
3.4.1. Pengukuran Kecepatan Putar Motor	43
3.4.2. Pengukuran Gaya Angkat <i>Propeller</i>	43
3.4.3. Pengukuran Hubungan <i>Throttle</i> Pada <i>Remote Control</i> dan Kecepatan Motor	43
3.5. Identifikasi dan Validasi Sistem	44
3.6. Model Matematika Hasil Identifikasi Sistem	47
3.7. Perancangan Kontroler	47
3.7.1. Pengaturan Posisi Sumbu Z	50
3.7.2. Pengaturan Sudut <i>Roll</i>	54
3.7.3. Pengaturan Sudut <i>Pitch</i>	57
3.7.4. Pengaturan Sudut <i>Yaw</i>	59
3.7.5. Pengaturan Posisi Sumbu X	61
3.7.6. Pengaturan Posisi Sumbu Y	63
3.8. Konversi Sinyal Kontrol Sumbu X dan Y ke Dalam Sudut Referensi	64
BAB IV HASIL DAN ANALISA	65
4.1. Pengujian Sensor	65
4.2. Simulasi	68
4.2.1. Respon <i>Open Loop</i>	68
4.2.2. Simulasi <i>Sliding Mode Control</i> dengan Fungsi Signum untuk Gerak <i>Waypoint Tracking</i>	69
4.2.3. Simulasi <i>Sliding Mode Control</i> dengan Fungsi Saturasi untuk Gerak <i>Waypoint Tracking</i>	74
4.2.4. Simulasi <i>Sliding Mode Control</i> dengan Struktur PID untuk Gerak <i>Waypoint Tracking</i>	77
4.2.5. Simulasi <i>Sliding Mode Control</i> dengan <i>Noise</i>	82

4.3. Simulasi 3D untuk Gerak <i>Waypoint Tracking</i> pada <i>Quadcopter</i>	86
BAB V PENUTUP	89
5.1. Kesimpulan	89
5.2. Saran	89
DAFTAR PUSTAKA	91
LAMPIRAN A. Pengukuran Kecepatan Motor	A1
LAMPIRAN B. Penghitungan Konstanta <i>Thrust</i> dan <i>Drag</i>	A3
LAMPIRAN C. Identifikasi Fisik Konstanta	A5
LAMPIRAN D. Blok Diagram Simulink	A9
RIWAYAT PENULIS	A17

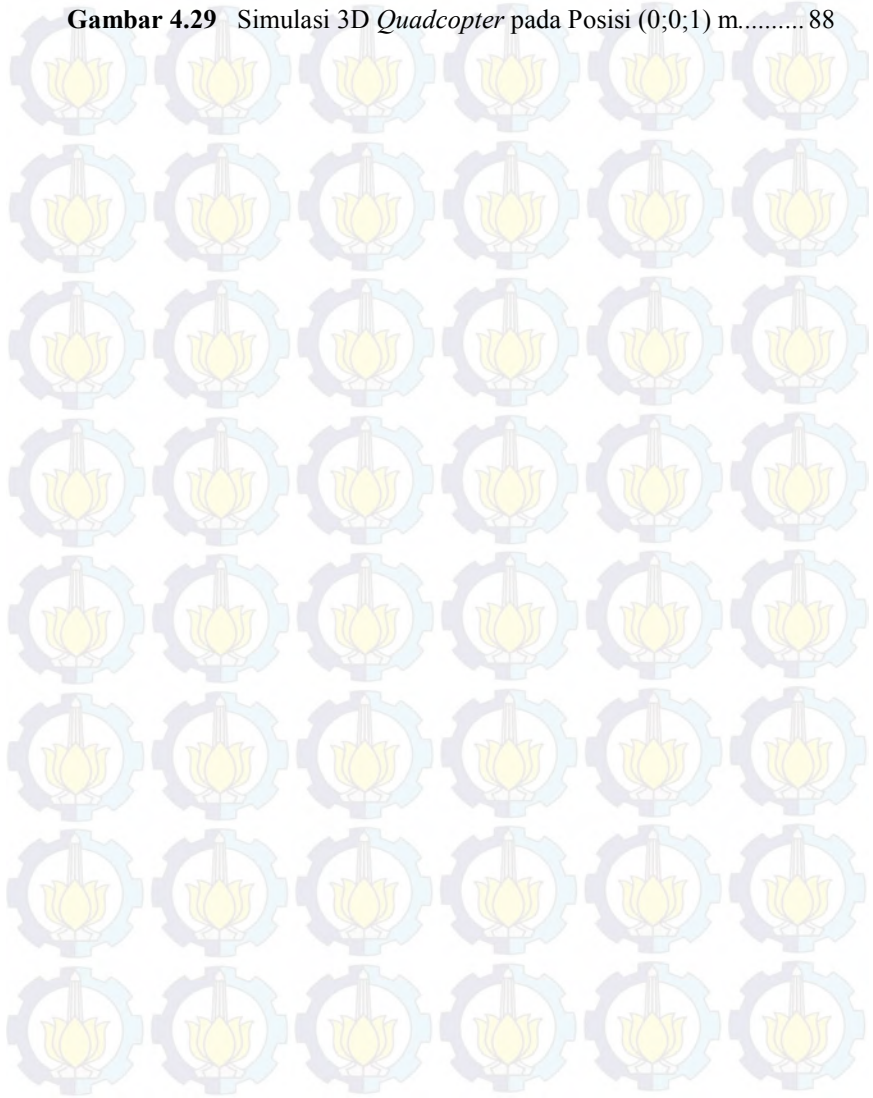
DAFTAR GAMBAR

Gambar 2.1	Gambar Umum <i>Quadcopter</i> dengan Arah Gerak <i>Roll</i> , <i>Pitch</i> , <i>Yaw</i>	8
Gambar 2.2	Gaya <i>Thrust</i>	9
Gambar 2.3	Gerak <i>Roll</i>	9
Gambar 2.4	Gerak <i>Pitch</i>	10
Gambar 2.5	Gerak <i>Yaw</i>	10
Gambar 2.6	Rotasi di Sekitar Sumbu Z_E oleh Sudut ψ (<i>Yaw</i>)	12
Gambar 2.7	Rotasi di Sekitar Sumbu Y_1 oleh Sudut θ (<i>Pitch</i>)	12
Gambar 2.8	Rotasi di Sekitar Sumbu x_2 oleh Sudut ϕ (<i>Roll</i>)	12
Gambar 2.9	<i>E-frame</i> dan <i>B-frame Quadcopter</i> beserta Empat Gaya Di Setiap <i>Propeller</i>	18
Gambar 2.10	APM 2.6	27
Gambar 2.11	<i>Transmitter</i> dan <i>Receiver</i>	28
Gambar 2.12	<i>Electronic Speed Controller</i>	29
Gambar 2.13	<i>Motor DC</i> dan <i>Propeller</i>	29
Gambar 2.14	<i>Ground Station</i>	30
Gambar 2.15	Prinsip Proporsional.....	31
Gambar 2.16	Karakteristik Kontroler Proporsional.....	32
Gambar 2.17	Prediksi Kesalahan.....	33
Gambar 2.18	Jalur (Garis yang Diberi Titik) dari Lintasan <i>State</i> dari Suatu Sistem dengan Kontroler Saat Diberi Gangguan Kecil	38
Gambar 2.19	Penggambaran <i>Phase-plane</i> dari Sistem yang Dikontrol dengan Menggunakan SMC pada <i>Switching Surface</i> .	38
Gambar 3.1	Desain Mekanik <i>Quadcopter</i> yang Dibuat	42
Gambar 3.2	Rancangan Sistem Elektronika <i>Quadcopter</i>	42
Gambar 3.3	Diagram Blok Pengaturan Ketinggian <i>Quadcopter</i>	50
Gambar 3.4	Permukaan Luncur.....	51
Gambar 3.5	Diagram Blok Pengaturan Sudut <i>Roll</i>	54
Gambar 3.6	Diagram Blok Desain Kontroler Pengaturan Sudut <i>Roll</i>	55
Gambar 3.7	Diagram Blok Desain Kontroler Pengaturan Sudut <i>Roll</i> (<i>Loop Luar</i>).....	56
Gambar 3.8	Diagram Blok Pengaturan Sudut <i>Pitch</i>	57
Gambar 3.9	Diagram Blok Pengaturan Sudut <i>Yaw</i>	59
Gambar 3.10	Diagram Blok Pengaturan Posisi <i>Quadcopter</i>	61

Gambar 4.1	Pengujian Sudut <i>Roll</i>	66
Gambar 4.2	Pengujian Sudut <i>Pitch</i>	67
Gambar 4.3	Pengujian Sudut <i>Yaw</i>	68
Gambar 4.4	Respon Posisi <i>Quadcopter</i> Sumbu Z tanpa Kontroler	69
Gambar 4.5	Respon Posisi <i>Quadcopter</i> Sumbu X tanpa Kontroler	69
Gambar 4.6	Respon Posisi <i>Quadcopter</i> Sumbu Y tanpa Kontroler	69
Gambar 4.7	Respon <i>Tracking Quadcopter</i> Sumbu X dengan SMC Signum.....	70
Gambar 4.8	Respon <i>Tracking Quadcopter</i> Sumbu Y dengan SMC Signum.....	71
Gambar 4.9	Lintasan <i>Tracking Quadcopter</i> dengan SMC Signum	71
Gambar 4.10	Sinyal Kontrol U_x dan U_y dengan SMC Signum	72
Gambar 4.11	<i>Sliding Surface</i> Gerak Sumbu X dengan SMC Signum	73
Gambar 4.12	Respon <i>Tracking Quadcopter</i> Sumbu X dengan SMC Saturasi	74
Gambar 4.13	Respon <i>Tracking Quadcopter</i> Sumbu Y dengan SMC Saturasi	75
Gambar 4.14	Lintasan <i>Tracking Quadcopter</i> dengan SMC Saturasi	75
Gambar 4.15	Sinyal Kontrol U_x dan U_y dengan SMC Saturasi.....	76
Gambar 4.16	<i>Sliding Surface</i> Gerak Sumbu X dengan SMC Saturasi	77
Gambar 4.17	Respon <i>Tracking Quadcopter</i> Sumbu X dengan PID- SMC.....	78
Gambar 4.18	Respon <i>Tracking Quadcopter</i> Sumbu Y dengan PID- SMC.....	79
Gambar 4.19	Lintasan <i>Tracking Quadcopter</i> dengan PID-SMC.....	79
Gambar 4.20	Sinyal Kontrol U_x dan U_y dengan PID-SMC.....	81
Gambar 4.21	<i>Sliding Surface</i> Gerak Sumbu X dengan PID-SMC ...	82
Gambar 4.22	Lintasan <i>Tracking Quadcopter</i> dengan <i>Noise</i>	83
Gambar 4.23	Respon Gerak Sumbu X <i>Quadcopter</i> dengan <i>Noise</i> ...	84
Gambar 4.24	Respon Gerak Sumbu Y <i>Quadcopter</i> dengan <i>Noise</i> ...	84
Gambar 4.25	Sinyal Kontrol U_x dan U_y pada Gerak Sumbu X dengan <i>Noise</i>	85
Gambar 4.26	Sinyal Kontrol U_x dan U_y pada Gerak Sumbu Y dengan <i>Noise</i>	86
Gambar 4.27	Simulasi 3D <i>Quadcopter</i> pada Posisi (0;0;0,99949) m.	87

Gambar 4.28 Simulasi 3D *Quadcopter* pada Posisi (1,023;0,99946;1) m..... 87

Gambar 4.29 Simulasi 3D *Quadcopter* pada Posisi (0;0;1) m..... 88



DAFTAR TABEL

Tabel 3.1	Parameter <i>Quadcopter</i>	45
Tabel 3.2	Konstanta Hasil Identifikasi.....	46
Tabel 3.3	Perbandingan Hasil Identifikasi.....	47
Tabel 4.1	Pengujian Sudut <i>Roll</i>	65
Tabel 4.2	Pengujian Sudut <i>Pitch</i>	66
Tabel 4.3	Pengujian Sudut <i>Yaw</i>	67
Tabel 4.4	<i>Error Tracking</i> Quadcopter dengan SMC Signum.....	72
Tabel 4.5	<i>Error Tracking</i> Quadcopter dengan SMC Saturasi.....	76
Tabel 4.6	<i>Error Tracking</i> Quadcopter dengan PID-SMC.....	80
Tabel 4.7	<i>Error Tracking</i> Quadcopter dengan <i>Noise</i> pada Sinyal Kontrol.....	83



BAB I

PENDAHULUAN

Tugas Akhir merupakan penelitian yang dilakukan oleh mahasiswa S1 Institut Teknologi Sepuluh Nopember Surabaya. Tugas Akhir ini merupakan salah satu syarat wajib untuk menyelesaikan studi dalam program sarjana teknik.

Pada bab ini, akan dibahas mengenai hal-hal yang mendahului pelaksanaan Tugas Akhir. Hal-hal tersebut meliputi latar belakang, perumusan masalah, tujuan, batasan masalah, sistematika penulisan, dan relevansi.

1.1. Latar Belakang

Unmanned Aerial Vehicle (UAV) atau pesawat terbang tanpa awak, memiliki peranan aplikasi yang semakin berkembang, di antaranya dalam bidang militer, ilmu kebumihantaran untuk mengambil citra suatu wilayah, dan bidang lainnya. Pesawat tanpa awak secara garis besar terbagi menjadi dua yaitu, *fixed wing* dan *rotorcraft*. Pesawat *fixed wing* memanfaatkan gerak translasi, sehingga dihasilkan perbedaan tekanan pada sayap pesawat bagian bawah dan atas, yang digunakan sebagai gaya angkat. Tekanan yang dihasilkan proporsional dengan perubahan kuadrat kecepatan angin yang mengalir di atas dan bawah pesawat. Untuk memperoleh tekanan yang besar dan mampu mengangkat pesawat, dibutuhkan kecepatan yang besar, dan mencapai kecepatan translasi yang besar akan membutuhkan jarak tempuh yang cukup panjang dan halus untuk mengurangi friksi yang terjadi. Selain itu, untuk mempertahankan ketinggian, pesawat *fixed wing* juga membutuhkan kecepatan yang cukup besar. Pesawat *rotorcraft* bergerak dengan rotor yang berputar dan menghasilkan gaya translasi vertikal akibat propulsi aliran udara. Dengan adanya gerak translasi vertikal ini memungkinkan pesawat memiliki kemampuan *take off* dan *landing* secara vertikal atau biasa disebut VTOL (*Vertical Take Off-Landing*).

Oleh karena itu, landasan pacu yang dibutuhkan oleh pesawat *rotorcraft* tidak harus memiliki jarak yang panjang dan juga memungkinkan untuk melakukan *take off-landing* pada landasan yang kasar bahkan miring. Jika pesawat *fixed-wing* membutuhkan kecepatan maju yang cukup besar untuk mempertahankan ketinggian, pesawat *rotorcraft* tidak memerlukan hal itu. Untuk pengambilan citra/informasi

yang detail, hal tersebut menjadi keuntungan penggunaan pesawat *rotorcraft* dibandingkan dengan pesawat *fixed-wing*. Pesawat *rotorcraft* yang paling konvensional adalah helikopter. Gerak maju atau menyamping dari helikopter dengan memiringkan *propeller* dengan sudut tertentu. Posisi *propeller* yang tidak bersifat kaku/konstan tersebut, menyebabkan helikopter sulit untuk dikendalikan.

Dengan membuat *rotor* pada pesawat *rotorcraft* menjadi empat buah, diharapkan dapat menggantikan kondisi yang memaksa untuk mengendalikan posisi sudut *propeller*. Pengaturan perubahan sudut posisi *propeller* yang digunakan oleh helikopter untuk bergerak maju atau menyamping dapat digantikan dengan hanya mengatur kecepatan dari keempat rotor pada *propeller*. Pesawat *rotorcraft* dengan empat buah *propeller* ini disebut juga *Quadro-helicopter*, *Quadcopter*, atau Quadrotor. Salah satu metode pengaturan posisi, baik secara translasi maupun rotasi yang dapat diaplikasikan pada plant ini adalah metode *Sliding Mode Control* yang merupakan salah satu teknik kontrol yang dapat digunakan pada plant yang nonlinear. Prinsip dasar untuk *Sliding Mode Control* terdiri dari perpindahan atau pergerakan keadaan lintasan ke arah bidang geser (*sliding surface*) dan mempertahankannya sekitar bidang ini dengan sebuah fungsi geser.

Namun, metode *Sliding Mode Control* ini mempunyai kelemahan, yaitu timbulnya *chattering* atau *switching* dengan frekuensi tinggi pada sinyal kontrol, yang disebabkan oleh sifat diskontinyu pada sinyal kontrol natural dari sistem. Fenomena *chattering* dapat merusak aktuator dari sistem yang digunakan. Oleh karena itu, dalam penelitian kali ini akan digunakan *Sliding Mode Control* berbasis PID, untuk menghilangkan efek *chattering* pada sinyal kontrol.

Perancangan kontroler akan disimulasikan dengan menggunakan software MATLAB. Pada implementasinya, penggunaan *Sliding Mode Control* berbasis PID diharapkan dapat menjaga posisi *quadcopter* saat melakukan gerakan *waypoint tracking* agar sesuai dengan *set point* yang diinginkan..

1.2. Permasalahan

Quadcopter diharapkan dapat bergerak dengan akurat dan mampu mengikuti referensi yang ditentukan. *Quadcopter* juga diharapkan memiliki pergerakan yang lincah, yang ditandai dengan kemampuan *quadcopter* untuk bergerak translasi maju dan menyamping, serta bergerak rotasi pada sumbu horizontal maupun vertikal dengan batasan

sudut tertentu. Secara fisis, pergerakan maju atau menyamping dari *quadcopter* membutuhkan sudut dan gaya angkat. Oleh karena itu, sudut dari *quadcopter* akan menjadi parameter untuk pergerakan translasi.

Pada pengaturan gerak translasi pada *quadcopter* dibutuhkan kontroler yang mampu beradaptasi dengan perubahan parameter. Gerak translasi yang akan dikontrol pada penelitian ini adalah gerak naik dan pendaratan (vertikal) dari *quadcopter*, serta gerak dengan arah sumbu x dan sumbu y. Untuk melakukan gerak *take off* dari *quadcopter*, mungkin hanya dibutuhkan peningkatan kecepatan *propeller* atau akselerasi positif vertikal. Begitu pula dengan gerakan ke arah sumbu x dan sumbu y, dapat diperoleh apabila kecepatan beberapa *propellernya* ditambah. Namun hal yang harus menjadi konsentrasi pada adalah kemampuan *Quadcopter* mencapai titik tertentu atau ketinggian tertentu dan dapat melakukan *tracking* mengikuti sinyal referensi. Oleh karena itu, diperlukan metode kontrol yang dapat membuat *plant* mencapai titik yang diinginkan, meskipun *quadcopter* mengalami gangguan-gangguan yang berasal dari luar. Dalam proses *tracking* posisi, *quadcopter* harus dapat meminimalkan error akibat berbagai gangguan yang ada, misalkan gangguan angin atau perubahan parameter. Selain itu dalam proses pendaratan *quadcopter* pun perlu diperhatikan adanya tumbukan antara landasan dan *quadcopter*. Untuk itu diperlukan kontrol gerak sehingga kecepatan vertikal *quadcopter* sebelum mendarat tidak terlalu besar.

1.3. Tujuan

Tujuan Tugas Akhir ini adalah merancang kontroler dengan metode *Sliding Mode Control* berbasis PID untuk pengaturan gerak *waypoint tracking* pada *plant quadcopter*, dan mensimulasikan kontroler tersebut pada *quadcopter* dengan MATLAB/ Simulink.

1.4. Batasan Masalah

Pada Tugas Akhir ini, parameter dari *quadcopter* yang digunakan adalah *quadcopter* Qb yang berada pada Laboratorium Teknik Pengaturan B-105, Jurusan Teknik Elektro ITS. Konfigurasi *propeller quadcopter* yang digunakan adalah berbentuk *plus (+)*.

Spesifikasi *quadcopter* yang digunakan adalah sebagai berikut :

- a) Kontroler Ardupilot Mega 2.6,
- b) *Frame* Talon Turnigy 2.0,
- c) *Transmitter* dan *Receiver* Radio FrSky V8FR-II 2.4Ghz,
- d) *Electronic Speed Controller* TBS Bulletproof,

- e) BLDC Motor *SunnySky X2212 KV980*,
- f) *Remote Control* Turnigy 9XR.

Pada penelitian ini, dilakukan pengendalian gerakan, *tracking* pada *quadcopter*. Akan ada 6 variabel yang akan dikontrol, yaitu posisi *quadcopter* pada sumbu X, posisi *quadcopter* pada sumbu Y, posisi *quadcopter* pada sumbu Z, serta sudut *roll*, *pitch*, dan *yaw* dari *quadcopter*. Pada pengaturan posisi *quadcopter* di sumbu X, Y, dan Z, digunakan kontroler *Sliding Mode* dengan sinyal kontrol natural atau U_N berupa struktur PID. Sedangkan, untuk pengaturan sudut *roll*, *pitch*, dan *yaw*, digunakan kontroler P untuk mengatur kecepatan sudut, dan kontroler PD untuk mengatur posisi. Kedua kontroler ini disambungkan secara *cascade*, dengan *outer loop* adalah kontroler posisi, dan *inner loop* adalah kontroler kecepatan.

1.5. Metodologi

Untuk meneliti Tugas Akhir ini, dilakukan beberapa tahapan dan metodologi, di antaranya adalah sebagai berikut:

a) Studi Literatur

Studi literatur berguna untuk mencari informasi atau data mengenai plant, kontroler, atau sistem secara keseluruhan. Studi literatur diperlukan sebagai landasan dalam mengerjakan Tugas Akhir agar diperoleh teori penunjang yang memadai, baik ilmu dasar, analisis, maupun metode penelitian. Hal ini dapat dilakukan dengan melihat acuan dari jurnal, buku teks, internet, dan lain-lain. Dengan adanya studi literatur, penelitian dapat dilakukan berdasarkan teori-teori yang telah ada sebelumnya.

b) Perancangan *Hardware* dan *Software*

Perancangan *hardware* dan *software* dipersiapkan untuk simulasi dan implementasi sistem. *Hardware* dan *software* tersebut berupa aktuator-aktuator dari sistem, kontroler, sensor, dan perangkat komunikasi untuk pengambilan data dan monitoring.

c) Pemodelan dan Identifikasi *Plant*

Pemodelan dan identifikasi *plant* merupakan salah satu tahap yang sangat penting dalam pengerjaan Tugas Akhir ini. Pemodelan dan identifikasi *plant quadcopter* dilakukan dengan identifikasi parametrik berupa pengukuran *input - output*, sehingga diperoleh parameter-parameter yang ada pada *quadcopter* untuk memperoleh model matematikanya. Dengan didapatkannya model

matematika dari *quadcopter*, perancangan kontroler dapat dilakukan.

d) Desain Kontroler dan Simulasi

Setelah diperoleh model matematika *quadcopter*, akan dilakukan perancangan *Sliding Mode Control* berbasis PID sebagai algoritma kendali untuk gerak *waypoint tracking* pada *quadcopter*. Setelah diperoleh parameter kontroler, kemudian dilakukan tahap simulasi dengan menggunakan perangkat lunak MATLAB.

e) Penulisan Buku Tugas Akhir

Buku tugas akhir ditulis secara intensif bila proses pengujian telah selesai. Pada saat proses simulasi sedang berjalan, dilakukan eksplorasi bahan-bahan untuk penulisan buku Tugas Akhir dari jurnal-jurnal ilmiah dan buku.

1.6. Sistematika

Sistematika pembahasan Tugas Akhir ini terdiri atas lima bab, yaitu pendahuluan, teori penunjang, perancangan sistem, pengujian dan analisis, serta penutup.

BAB I PENDAHULUAN

Pada bab ini, dibahas mengenai latar belakang, perumusan permasalahan, tujuan, metodologi, sistematika penulisan, dan relevansi dari Tugas Akhir.

BAB II TEORI DASAR

Bab ini berisi tentang teori-teori yang berkaitan dengan topik yang diangkat dalam Tugas Akhir ini. Teori yang dibahas pada bab ini adalah mengenai dinamika dan kinematika *quadcopter*, analisis kestabilan Lyapunov, kontroler PID, dan *Sliding Mode Control*.

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perancangan algoritma kontrol *Waypoint Tracking* dengan *Sliding Mode Control* berbasis PID berdasarkan teori pada Bab II.

BAB IV HASIL DAN ANALISIS

Bab ini berisi tentang hasil pengujian simulasi kontrol *Waypoint Tracking* dengan *Sliding Mode Control* pada sistem, pada berbagai kondisi yang mendekati kondisi aktual.

BAB V PENUTUP

Setelah proses pengujian dilakukan, disajikan beberapa kesimpulan yang disertai dengan saran terhadap kelanjutan Tugas Akhir ini.

1.7. Relevansi

Hasil yang diperoleh pada Tugas Akhir ini diharapkan menjadi referensi desain kontroler pada *quadcopter*, pengembangan dan perbandingan metode kontrol yang tepat untuk pengaturan gerak pada *quadcopter*. Selain itu, juga untuk mengetahui pengaruh metode kontrol dengan *Sliding Mode Control* dengan berbasis PID pada sistem *quadcopter*.

BAB II

TEORI DASAR

Dalam penyusunan Tugas Akhir ini terdapat beberapa teori dasar yang menjadi acuan untuk merumuskan dan menyelesaikan masalah yang akan dibahas. Pada bagian awal diberikan tinjauan pustaka yang menggambarkan landasan teori secara umum yang akan digunakan pada tugas akhir ini. Pada bagian selanjutnya membahas tentang teori-teori pendukung, meliputi dinamika dan kinematika *quadcopter*, kontroler PID, *Sliding Mode Control*, dan analisa kestabilan Lyapunov.

2.1. Tinjauan Pustaka [1]

Unmanned Aerial Vehicle (UAV) berkembang pesat dalam beberapa tahun terakhir. Hal ini dikarenakan perannya pada berbagai bidang seperti dalam hal keamanan, eksplorasi lingkungan, agrikultural, militer, pemetaan kota, dan lain sebagainya. Untuk menunjang aplikasi-aplikasi UAV yang sangat luas, diperlukan kemampuan *Vertical Take-Off* dan *Landing* (VTOL), gerak *waypoint tracking*, dan gerak *hover* yang sesuai dengan kemampuan bergerak *quadcopter* yang cepat agar *quadcopter* menjadi kendaraan yang ideal.

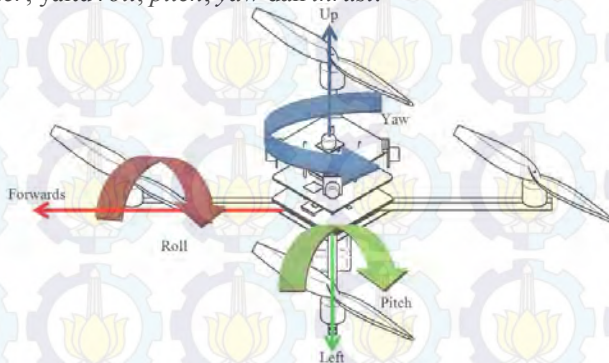
Setelah melalui banyak penelitian, tipe peralatan udara yang cocok untuk kondisi yang ada didalam ruangan dan diluar ruangan ialah *rotorcraft* atau *quadcopter helicopter*. *Quadcopter* merupakan peralatan udara yang digerakkan oleh 4 baling-baling atau *rotor*. *Quadcopter* dapat dikontrol dengan mengatur kecepatan pada setiap *rotornya* sehingga mengubah besarnya gaya angkat dan torsi yang dihasilkan setiap *rotor*.

Quadcopter diklasifikasikan sebagai sebuah *rotorcraft*, dan tidak termasuk dalam golongan *fixed-wing*, sebab daya angkatnya diperoleh dari empat *rotor*. Daya angkat dari *fixed-wing* diperoleh dari perbedaan aliran udara diantar bagian atas sayap dan bagian bawah sayap. Penggunaan empat *rotor* mengakibatkan setiap *rotornya* memiliki diameter yang lebih kecil jika dibandingkan dengan *helicopter* dengan satu *rotor*. Penggunaan empat *rotor* ini akan mengurangi energi kinetik yang dihasilkan selama pesawat terbang, karena energi kinetik dipengaruhi oleh massa *rotor* dan kecepatannya. Bentuk *rotor* yang lebih kecil akan mengurangi resiko adanya kerusakan yang disebabkan baling-baling mengenai obyek tertentu. *Rotor* dilindungi dari tabrakan yang keras dengan sebuah *frame* pelindung.

2.1.1 Konsep Dasar [2]

Quadcopter merupakan bagian dari *multirotor* untuk kategori pesawat *rotorcraft* yang mampu lepas landas dan mendarat secara vertikal. Perilaku terbang dari *quadcopter* bisa dikendalikan dengan mengatur kecepatan *propeller* yang berputar di keempat sisi, yang secara tidak langsung juga mengatur gaya angkat pada setiap sisinya. Bila dibandingkan dengan helikopter, *quadcopter* memiliki kelebihan di antaranya tidak memerlukan ekor penyeimbang seperti helikopter dan bisa menggunakan *propeller* yang *fixed pitch* atau tidak perlu mengatur sudut *propeller* seperti pada helikopter. *Quadcopter* yang digunakan sebagai robot terbang kecil memiliki model mekanik yang terdiri dari empat *rotor* yang dipasang pada sumbu silang simetris. Bentuk ini diharapkan tipis dan kaku, sehingga diperoleh friksi udara yang kecil dan komponen yang bergerak pada *quadcopter* hanyalah putaran *propeller*.

Setiap *propeller* pada *quadcopter* ini diputar oleh satu motor elektrik, sehingga terdapat empat motor sebagai aktuatur untuk menghasilkan gaya angkat dari *quadcopter*. Dengan batasan menggunakan karakteristik motor dan *propeller* yang relatif sama, maka kondisi melayang yang stabil akan diperoleh kecepatan motor yang sama di tiap *propeller*. Namun pada tiap *propeller* yang berputar akan menghasilkan torsi yang bergerak bidang horizontal, maka konfigurasi *propeller* terdiri dari dua yang bergerak searah jarum jam dan dua yang berlawanan arah jarum jam dan dalam tiap satu sumbu, *propeller* berputar dengan arah yang sama. Terdapat empat *input* yang terdapat pada *quadcopter*, yaitu *roll*, *pitch*, *yaw* dan *thrust*.

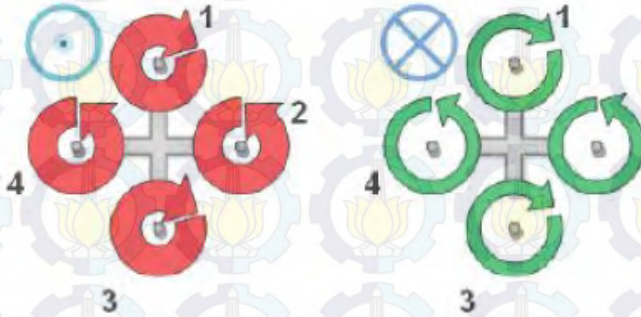


Gambar 2.1 Gambar Umum *Quadcopter* dengan Arah Gerak *Roll*, *Pitch*, *Yaw*

Dengan melakukan pengaturan kecepatan putaran *propeller* akan dihasilkan beberapa perintah *input*. di antaranya:

a) Gerak *Thrust*

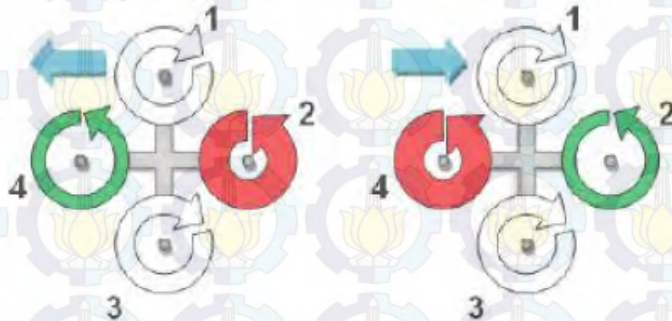
Dengan mempercepat (warna merah) atau memperlambat (warna hijau) kecepatan motor dari *propeller* secara bersamaan akan menghasilkan percepatan vertikal.



Gambar 2.2 Gaya *Thrust*

b) Gerak *Roll*

Dengan mempercepat atau memperlambat *propeller* pada sisi kiri dan secara bersamaan memperlambat atau mempercepat *propeller* di sisi kanan akan diperoleh rotasi dengan percepatan sudut *roll*.



Gambar 2.3 Gerak *Roll*

c) Gerak *Pitch*

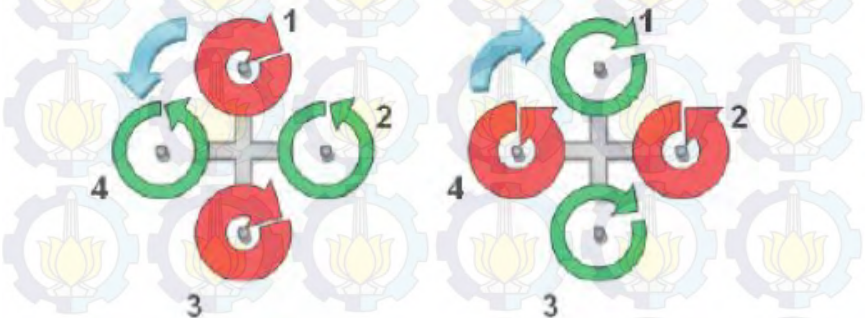
Gerak *Pitch* memiliki mekanisme yang sama seperti gerak *roll*, namun dilakukan pada *propeller* di sisi depan dan belakang akan diperoleh percepatan sudut *pitch*.



Gambar 2.4 Gerak *Pitch*

d) Gerak *Yaw*

Dengan mempercepat atau memperlambat *propeller* yang berputar berlawanan arah jarum jam, dan memperlambat atau mempercepat *propeller* yang berputar searah jarum jam akan diperoleh gerak *yaw*.



Gambar 2.5 Gerak *Yaw*

2.1.2 Kinematika *Quadcopter*

Pada bagian ini, akan dipelajari model dinamik *Quadcopter* UAV secara umum. Sistem yang relatif ringan dan mampu terbang diudara,

bentuk dinamik idealnya meliputi *gyroscopic effect* yang merupakan hasil resultan dari rotasi *rigid-body* pesawat diangkasa dan rotasi keempat *propellernya*. Model dinamik UAV menggunakan formulasi Euler-Lagrange. *Quadcopter* yang digunakan mempunyai konfigurasi *propeller* berbentuk *plus (+)*.

Quadcopter memiliki 6 *degree of freedom (DOF)*. Untuk mendeskripsikan gerakan dari 6 *DOF rigid-body* digunakan dua buah frame referensi yaitu *earth inertial reference (E-frame)* dan *body fixed reference (B-frame)*. *B-frame* dan *E-frame* diasumsikan berada pada pusat gravitasi dari *Quadcopter* UAV, di mana sumbu z mengarah ke atas, seperti terlihat pada gambar 2.10.

Persamaan gerak lebih mudah diformulasikan dalam *B-frame* karena matriks inersianya bersifat tidak berubah terhadap waktu, persamaan bisa disederhanakan dengan memanfaatkan keuntungan dari badan *quadcopter* yang simetris, pengukuran lebih mudah dikonversikan kedalam *B-frame*, dan pengaturan gaya selalu diberikan pada *Body fixed frame*.

E-frame (O_E , X_E , Y_E , Z_E) dipilih sebagai acuan inersia benda. Y_E menunjuk ke utara, X_E menunjuk ke timur, Z_E menunjuk ke atas (menjauhi bumi) dan O_E menunjuk ke titik asal. *Frame* ini digunakan untuk menentukan posisi linear Γ_E (dalam meter) dan posisi angular Θ_E (dalam radian) dari *quadcopter*.

B-frame menempel pada tubuh *quadcopter*. X menunjuk pada bagian depan *Quadcopter*, y menunjuk pada bagian kiri *quadcopter*, z menunjuk pada bagian atas, o menunjuk pada titik asal atau *origin*. Titik asal o digunakan untuk menunjukkan pusat dari *quadcopter*. Didalam *frame* ini ada kecepatan linear V_B (m/s), kecepatan angular Ω_B (rad/s), gaya F_B (N) dan torsi τ_B (Nm).

Posisi linear *quadcopter* (Γ_E) ditentukan dari koordinat vektor antara origin dan *B-frame* serta origin dari *E-frame* dengan memperhatikan *E-frame* dapat dilihat pada Persamaan (2.1).

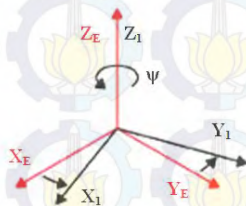
$$\Gamma^E = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \quad (2.1)$$

Posisi angular *quadcopter* (Θ_E) ditentukan dari orientasi *B-frame* terhadap *E-frame*. Hal ini diperoleh dari tiga bentuk rotasi yang berurutan pada sumbu utama yang mengambil *E-frame* menjadi *B-frame*. Ketiga

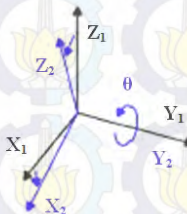
rotasi tersebut adalah *roll*, *pitch* dan *yaw*. Untuk mencari persamaan ketiga bentuk rotasi tersebut digunakan *Euler Model*.

$$\Theta^E = [\phi \quad \theta \quad \psi]^T \quad (2.2)$$

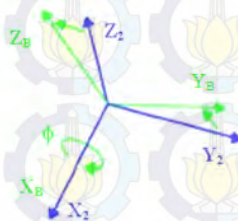
Matriks rotasi R_Θ diperoleh dari perkalian tiga rotasi matriks, seperti pada Gambar 2.11, Gambar 2.12 dan Gambar 2.13.



Gambar 2.6 Rotasi di Sekitar Sumbu Z_1 oleh Sudut ψ (*Yaw*)



Gambar 2.7 Rotasi di Sekitar Sumbu Y_1 oleh Sudut θ (*Pitch*)



Gambar 2.8 Rotasi di Sekitar Sumbu X_2 oleh Sudut ϕ (*Roll*)

Rotasi di sekitar sumbu z_E oleh sudut ψ (*yaw*) dengan matriks rotasi $R(\psi)$, ditunjukkan pada Gambar 2.6.

$$\begin{cases} x_E = x_1 \cos \psi - y_1 \sin \psi \\ y_E = y_1 \cos \psi + x_1 \sin \psi \\ z_E = z_1 \end{cases} \quad (2.3)$$

Persamaan (2.3) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.4).

$$\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (2.4)$$

Persamaan matriks $\mathbf{R}(\psi)$ adalah seperti pada Persamaan (2.5).

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Rotasi di sekitar sumbu y_1 oleh sudut θ (*pitch*) dengan matriks rotasi $\mathbf{R}(\theta)$, ditunjukkan pada Gambar 2.7.

$$\begin{cases} x_1 = x_2 \cos \theta - z_2 \sin \theta \\ y_1 = y_2 \\ z_1 = z_2 \cos \theta + x_2 \sin \theta \end{cases} \quad (2.6)$$

Persamaan (2.6) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.7).

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (2.7)$$

Persamaan matriks $\mathbf{R}(\theta)$ ditunjukkan pada Persamaan (2.8).

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.8)$$

Rotasi di sekitar sumbu x_2 oleh sudut ϕ (*roll*) dengan matriks rotasi $\mathbf{R}(\phi)$, ditunjukkan pada Gambar 2.8.

$$\begin{cases} x_2 = x_B \\ y_2 = y_B \cos \phi - z_B \sin \phi \\ z_2 = z_B \cos \phi + y_B \sin \phi \end{cases} \quad (2.9)$$

Persamaan (2.9) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.10).

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \sin \phi \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad (2.10)$$

Persamaan matriks $\mathbf{R}(\phi)$ adalah seperti pada Persamaan (2.11).

$$\mathbf{R}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \sin \phi \end{bmatrix} \quad (2.11)$$

Dari ketiga persamaan diatas, diperoleh persamaan matriks untuk gerak rotasi seperti pada Persamaan (2.12).

$$\mathbf{R}_\Theta = \mathbf{R}(\phi)\mathbf{R}(\theta)\mathbf{R}(\psi) \quad (2.12)$$

$$\mathbf{R}_\Theta = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta s_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (2.13)$$

Didalam persamaan sebelum dan sesudah ini, ada beberapa notasi yaitu $c_k = \cos k$, $s_k = \sin k$, $t_k = \tan k$. Seperti yang telah dinyatakan sebelumnya, kecepatan linear \mathbf{V}_B dan angular $\boldsymbol{\omega}_B$ diperlihatkan oleh *body-fixed frame* (B-frame). Komposisi vektornya ditunjukkan pada Persamaan (2.14) dan (2.15).

$$\mathbf{V}^B = [u \quad v \quad w]^T \quad (2.14)$$

$$\boldsymbol{\omega}^B = [p \quad q \quad r]^T \quad (2.15)$$

Diperlukan kombinasi nilai linear dan angular untuk memberikan representasi yang lengkap dalam *space*. ξ merupakan komposisi dari vektor posisi linear $\boldsymbol{\Gamma}^E$ (m) dan vektor posisi sudut $\boldsymbol{\Theta}^E$ (rad) terhadap E-frame seperti terlihat pada Persamaan (2.16).

$$\xi = [\boldsymbol{\Gamma}^E \quad \boldsymbol{\Theta}^E]^T = [X \quad Y \quad Z \quad \phi \quad \theta \quad \psi]^T \quad (2.16)$$

\mathbf{V} merupakan generalisasi dari vektor kecepatan linear *quadcopter* \mathbf{V}^B (m s^{-1}) dan kecepatan angular *quadcopter* $\boldsymbol{\omega}^B$ (rad s^{-1}) pada B-frame seperti dinyatakan pada Persamaan (2.17).

$$\mathbf{V} = [\mathbf{V}^B \quad \boldsymbol{\omega}^B]^T = [u \quad v \quad w \quad p \quad q \quad r]^T \quad (2.17)$$

Hubungan antara kecepatan linear pada B-frame dan salah satu faktor pada E-frame \mathbf{V}_E (atau $\boldsymbol{\Gamma}_E$) menyebabkan matriks rotasi \mathbf{R}_θ menurut Persamaan (2.18).

$$\mathbf{V}^E = \dot{\boldsymbol{\Gamma}}^E = \mathbf{R}_\theta \mathbf{V}^B \quad (2.18)$$

Seperti pada kecepatan linear, hal tersebut juga berlaku untuk menghubungkan kecepatan angular pada E-frame (atau kecepatan Euler) $\boldsymbol{\Theta}_E$ [rad/s] ke B-frame $\boldsymbol{\omega}_B$ berkat matriks transformasi \mathbf{T}_θ . Persamaan (2.18) dan (2.19) menunjukkan hubungan tersebut.

$$\boldsymbol{\omega}^B = \mathbf{T}_\theta^{-1} \dot{\boldsymbol{\Theta}}^E \quad (2.19)$$

$$\dot{\boldsymbol{\Theta}}^E = \mathbf{T}_\theta \boldsymbol{\omega}^B \quad (2.20)$$

Matriks transformasi T_θ dapat ditetapkan dengan menggunakan kecepatan Euler dalam *B-frame*, dengan membalik pola perputaran sudut dari *roll*, *pitch* dan *yaw* seperti pada Persamaan (2.21).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi)^{-1} R(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_\theta^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.22)$$

Dari Persamaan (2.13) dan (2.19) maka diperoleh matriks transformasi dari bingkai bodi menuju bingkai bumi.

$$T_\theta^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \quad (2.23)$$

$$T_\theta = \begin{bmatrix} 1 & t_\theta s_\phi & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \quad (2.24)$$

2.1.3 Dinamika *Quadcopter*

Dinamika adalah cabang dari ilmu fisika (terutama mekanika klasik) yang mempelajari gaya dan torsi dan efeknya pada gerak. Dinamika merupakan kebalikan dari kinematika, yang mempelajari gerak suatu objek tanpa memperhatikan apa penyebabnya. Secara umum, dinamika sangat berkaitan erat dengan Hukum kedua Newton tentang gerak.

Berdasarkan aksioma pertama Euler dari Hukum Kedua Newton, turunan dari komponen linear dari gerakan suatu benda dapat dilihat pada Persamaan (2.25).

$$\left\{ \begin{array}{l} m \ddot{\Gamma}^E = F^E \\ m \overbrace{R_\Theta \dot{V}^B} = R_\Theta F^B \\ m (R_\Theta \dot{V}^B + \dot{R}_\Theta V^B) = R_\Theta F^B \\ m R_\Theta (\dot{V}^B + \omega^B \times V^B) = R_\Theta F^B \\ m (\dot{V}^B + \omega^B \times V^B) = F^B \end{array} \right. \quad (2.25)$$

Di mana m (kg) adalah massa *quadcopter*, $\ddot{\Gamma}^E$ ($m s^{-2}$) adalah vektor percepatan linear *quadcopter* yang mengacu pada E-frame, F^E (N) adalah vektor gaya *quadcopter* terhadap E-frame, \dot{V}^B ($m s^{-2}$) adalah percepatan linear *quadcopter* terhadap B-frame, dan \dot{R}_Θ adalah turunan pertama matriks rotasi. Dan simbol \times merupakan perkalian produk suatu vektor.

Berdasarkan aksioma kedua Euler dari Hukum Kedua Newton, dengan cara yang sama, turunan dari gerakan angular komponen dari suatu benda dapat dilihat pada Persamaan (2.26).

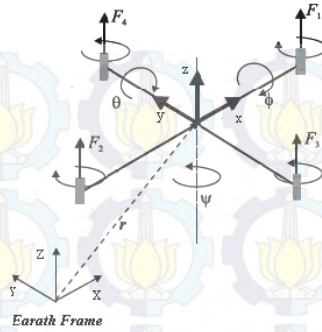
$$\left\{ \begin{array}{l} I \ddot{\Theta}^E = \tau^E \\ I \overbrace{T_\Theta \omega^B} = T_\Theta \tau^B \\ I \dot{\omega}^B + \omega^B \times (I \omega^B) = \tau^B \end{array} \right. \quad (2.26)$$

Pada Persamaan (2.26), I ($N m s^2$) adalah matriks inersia *quadcopter* (pada B-frame), $\ddot{\Theta}^E$ ($rad s^{-2}$) adalah vektor percepatan sudut *quadcopter* terhadap E-frame, $\dot{\omega}^B$ ($rad s^{-2}$) adalah vektor percepatan sudut *quadcopter* terhadap B-frame, dan τ^E (N m) adalah torsi *quadcopter* terhadap E-frame.

Dari Persamaan (2.25) dan (2.26), dapat dibentuk persamaan benda kaku dengan 6 derajat kebebasan (DOF). Persamaan (2.27) menunjukkan formulasi matriks dari dinamika sistem *quadcopter*.

$$\begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (m V^B) \\ \omega^B \times (I \omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad (2.27)$$

Di mana notasi $I_{3 \times 3}$ berarti matriks identitas 3×3 . Dan $0_{3 \times 3}$ adalah matriks nol 3×3 . *Quadcopter* memiliki empat buah motor yang menghasilkan gaya dorong (*thrust*) seperti pada Gambar 2.9.



Gambar 2.9 E-frame dan B-frame *Quadcopter* beserta empat gaya disetiap *propeller*

Vektor gaya yang terjadi pada *quadcopter* dapat ditentukan berdasarkan Persamaan (2.28).

$$\Lambda = [F^B \ \tau^B]^T = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \quad (2.28)$$

Persamaan (2.28) dapat juga ditulis dalam bentuk formulasi matriks seperti pada persamaan (2.29).

$$\mathbf{M}_B \dot{\mathbf{v}} + \mathbf{C}_B(\mathbf{v})\mathbf{v} = \Lambda \quad (2.29)$$

Di mana $\dot{\mathbf{v}}$ adalah vektor percepatan *quadcopter* terhadap B-frame. \mathbf{M}_B adalah matriks inersia sistem dan \mathbf{C}_B adalah matriks sentripetal Coriolis. Persamaan (2.30) menunjukkan matriks \mathbf{M}_B .

$$\mathbf{M}_B = \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (2.30)$$

Dapat dilihat bahwa matriks \mathbf{M}_B adalah matriks diagonal, sedangkan matriks \mathbf{C}_B ditunjukkan oleh Persamaan (2.31).

$$C_B = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -m \mathbf{S}(\mathbf{V}_B) \\ \mathbf{0}_{3 \times 3} & -\mathbf{S}(\mathbf{I} \boldsymbol{\omega}_B) \end{bmatrix}$$

$$C_B = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \quad (2.31)$$

Pada persamaan ini, diadopsi operator *skew-symmetric* $\mathbf{S}(\cdot)$. Untuk vektor 3 dimensi \mathbf{k} , matriks *skew-symmetric* dari \mathbf{k} ($\mathbf{S}(\mathbf{k})$) ditunjukkan pada Persamaan (2.32).

$$\mathbf{S}(\mathbf{k}) = -\mathbf{S}^T(\mathbf{k}) = \begin{bmatrix} 0 & -k_3 & k_1 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}, \mathbf{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (2.32)$$

Matriks $\mathbf{\Lambda}$ dapat dibagi menjadi tiga komponen menurut kontribusi sifat dasar *quadcopter*. Yang pertama adalah vektor gravitasi $\mathbf{G}_B(\boldsymbol{\xi})$ yang diperoleh dari percepatan gravitasi g (m s^{-2}). Percepatan gravitasi hanya berlaku pada persamaan linear *quadcopter*. Persamaan matriks $\mathbf{G}_B(\boldsymbol{\xi})$ dituliskan pada Persamaan (2.33).

$$\begin{aligned} \mathbf{G}_B(\boldsymbol{\xi}) &= \begin{bmatrix} F_{GB} \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\theta^{-1} F_{GE} \\ 0_{3 \times 1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_\theta^T \\ 0_{3 \times 1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = \begin{bmatrix} mg \sin \theta \\ -mg \cos \theta \sin \phi \\ -mg \cos \theta \sin \phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (2.33)$$

Di mana \mathbf{F}_{GB} (N) adalah vektor gaya gravitasi terhadap B-frame dan \mathbf{F}_{GE} (N) terhadap E-frame. \mathbf{R}_θ adalah matriks ortogonal, sehingga matriks inversnya sama dengan matriks transposnya.

Kontribusi yang kedua mengenai efek giroskopis yang dihasilkan oleh pergerakan *propeller*. Selama dua *propeller* yang berhadapan berputar searah jarum jam dan dua *propeller* lainnya berputar berlawanan arah jarum jam, terjadi ketidakseimbangan saat jumlah aljabar dari kecepatan motor tidak sama dengan nol. Jika *roll* dan *pitch* juga tidak nol, *quadcopter* akan mengalami torsi efek giroskopis yang ditunjukkan pada Persamaan (2.34).

$$\begin{aligned} \mathbf{O}_B(\mathbf{v})\boldsymbol{\Omega} &= \left[-\sum_{k=1}^4 J_{TP} \left(\mathbf{\omega}^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (-1)^k \boldsymbol{\Omega}_k \right] = \left[J_{TP} \begin{bmatrix} 0_{3 \times 1} \\ -q \\ p \\ 0 \end{bmatrix} \boldsymbol{\Omega} \right] \\ &= J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -q & -q & -q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega} \end{aligned} \quad (2.34)$$

\mathbf{O}_B adalah matriks giroskopis dari *propeller* dan J_{TP} (N m s^2) adalah momen inersia total di sekitar sumbu *propeller*. Efek giroskopis yang dihasilkan oleh putaran *propeller* hanya berhubungan dengan persamaan angular, dan tidak mempengaruhi persamaan linear. $\boldsymbol{\Omega}$ (rad s^{-1}) merupakan vektor kecepatan putar *propeller*, yang ditunjukkan oleh Persamaan (2.35).

$$\boldsymbol{\Omega} = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4, \quad \boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (2.35)$$

Di mana Ω_1 , Ω_2 , Ω_3 , dan Ω_4 (rad s^{-1}) adalah kecepatan *propeller* depan, kanan, belakang, dan kiri.

Kontribusi yang ketiga adalah perhitungan gaya dan torsi yang dihasilkan oleh pergerakan *input*. Pergerakan matriks \mathbf{E}_B dikali dengan $\boldsymbol{\Omega}^2$ untuk memperoleh vektor perpindahan $\mathbf{U}_B(\boldsymbol{\Omega})$. Efek aerodinamis (faktor *thrust* b (N s^2) dan *drag* d (N m s^2)) berpengaruh pada gaya dan

torsi yang dihasilkan. Persamaan vektor perpindahan pada dinamika *quadcopter* ditunjukkan pada Persamaan (2.36).

$$\mathbf{U}_B(\boldsymbol{\Omega}) = \mathbf{E}_B \boldsymbol{\Omega}^2 = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\boldsymbol{\Omega}_1^2 + \boldsymbol{\Omega}_2^2 + \boldsymbol{\Omega}_3^2 + \boldsymbol{\Omega}_4^2) \\ bl(-\boldsymbol{\Omega}_2^2 + \boldsymbol{\Omega}_4^2) \\ bl(-\boldsymbol{\Omega}_1^2 + \boldsymbol{\Omega}_3^2) \\ d(-\boldsymbol{\Omega}_1^2 + \boldsymbol{\Omega}_2^2 - \boldsymbol{\Omega}_3^2 + \boldsymbol{\Omega}_4^2) \end{bmatrix} \quad (2.36)$$

Di mana l (m) adalah jarak antara pusat *quadcopter* dengan pusat *propeller*. U_1 , U_2 , U_3 , dan U_4 adalah komponen vektor gerak. Hubungan komponen vektor tersebut dengan *propeller* dapat diperoleh dari perhitungan aerodinamis.

Dari persamaan sebelumnya, dapat diketahui konstanta matriks \mathbf{E}_B yang dikalikan dengan kuadrat dari kecepatan *propeller* $\boldsymbol{\Omega}^2$ sehingga menghasilkan $\mathbf{U}_B(\boldsymbol{\Omega})$. Persamaan (2.37) menunjukkan matriks \mathbf{E}_B .

$$\mathbf{E}_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \quad (2.37)$$

Dari Persamaan (2.29), dinamika *quadcopter* dapat dibuat dengan mempertimbangkan ketiga kontribusi menurut Persamaan (2.38).

$$\mathbf{M}_B \dot{\mathbf{v}} + \mathbf{C}_B(\mathbf{v})\mathbf{v} = \mathbf{G}_B(\boldsymbol{\xi}) + \mathbf{O}_B(\mathbf{v}) \boldsymbol{\Omega} + \mathbf{E}_B \boldsymbol{\Omega}^2 \quad (2.38)$$

Dengan mengatur ulang Persamaan (2.38) diatas, turunan dari vektor kecepatan terhadap B-frame dapat dirumuskan seperti pada Persamaan (2.39).

$$\dot{\mathbf{v}} = \mathbf{M}_B^{-1} (\mathbf{C}_B(\mathbf{v})\mathbf{v} + \mathbf{G}_B(\boldsymbol{\xi}) + \mathbf{O}_B(\mathbf{v}) \boldsymbol{\Omega} + \mathbf{E}_B \boldsymbol{\Omega}^2) \quad (2.39)$$

Persamaan (2.40) menunjukkan persamaan – persamaan sebelumnya dinyatakan bukan dalam bentuk matriks, melainkan dalam bentuk sistem persamaan.

$$\begin{cases} \dot{u} = (v r - w q) + g s_\theta \\ \dot{v} = (w p - u r) - g c_\theta s_\phi \\ \dot{w} = (u q - v p) - g c_\theta s_\phi + \frac{U_1}{m} \\ \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} q r - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} p r - \frac{J_{TP}}{I_{YY}} p \Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} p q - \frac{U_4}{I_{ZZ}} \end{cases} \quad (2.40)$$

Persamaan *input* kecepatan *propeller* (baling-baling) dapat dilihat dalam Persamaan (2.41).

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = bl(-\Omega_2^2 + \Omega_4^2) \\ U_3 = bl(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (2.41)$$

Persamaan sistem dinamik *quadcopter* (2.40) diperoleh dari *B-frame*. Dalam pemodelannya, digunakan persamaan dinamik sistem yang mencakup sistem *hybrid* yang terdiri dari persamaan linear dari *E-frame* dan persamaan angular dari *B-frame*. Hal ini dilakukan untuk mempermudah pengaturan. Persamaan-persamaan berikut akan disajikan dalam kerangka *hybrid*, atau *H-frame*. Persamaan (2.42) menunjukkan vektor kecepatan *quadcopter* yang digeneralisasi pada *H-frame*.

$$\dot{\xi} = [\Gamma^E \quad \omega^B]^T = [\dot{X} \quad \dot{Y} \quad \dot{Z} \quad p \quad q \quad r]^T \quad (2.42)$$

Dinamika sistem pada *H-frame* dapat dituliskan dalam bentuk matriks menurut Persamaan (2.43).

$$M_H \dot{\xi} + C_H(\xi)\xi = G_H + O_H(\xi)\Omega + E_H(\xi)\Omega^2 \quad (2.43)$$

Berikut ini akan ditentukan semua matriks dan vektor yang ditunjukkan oleh persamaan di atas. Matriks inersia sistem terhadap *H-*

frame \mathbf{M}_H sama dengan matriks inersia sistem terhadap B-*frame*, dan dinyatakan seperti pada Persamaan (2.44).

$$\mathbf{M}_H = \mathbf{M}_B = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (2.44)$$

Namun, matriks sentripetal Coriolis terhadap H-*frame* $\mathbf{C}_H(\boldsymbol{\zeta})$ tidak sama dengan matriks sentripetal Coriolis terhadap B-*frame* dan dinyatakan seperti pada Persamaan (2.45).

$$\mathbf{C}_H(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -S(I \boldsymbol{\omega}_B) \end{bmatrix} \quad (2.45)$$

Vektor gravitasi terhadap H-*frame* \mathbf{G}_H dituliskan pada Persamaan (2.46). Dapat dilihat bahwa gravitasi mempengaruhi ketiga persamaan linear, namun lebih berpengaruh terhadap ketinggian *quadcopter*.

$$\mathbf{G}_H = \begin{bmatrix} F_G^E \\ \mathbf{0}_{3 \times 1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.46)$$

Efek giroskopis yang dihasilkan oleh putaran *propeller* tidak berubah karena hanya mempengaruhi persamaan angular yang mengacu kepada B-*frame*. Oleh karena itu, matriks giroskopis terhadap H-*frame* $\mathbf{O}_H(\boldsymbol{\zeta})$ dibuat sama dengan Persamaan (2.47).

$$\begin{aligned}
\mathbf{O}_H(\xi)\boldsymbol{\Omega} &= \mathbf{O}_B(v)\boldsymbol{\Omega} = \begin{bmatrix} 0_{3 \times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \boldsymbol{\Omega} \end{bmatrix} \\
&= \mathbf{J}_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega}
\end{aligned} \tag{2.47}$$

Matriks perpindahan terhadap H-frame $\mathbf{E}_H(\xi)$ berbeda dengan perpindahan terhadap B-frame karena input U_1 mempengaruhi semua persamaan linear dengan matriks rotasi \mathbf{R}_θ . Perkalian produk antara matriks perpindahan dengan kuadrat kecepatan *propeller* ditunjukkan pada Persamaan (2.48).

$$\begin{aligned}
\mathbf{E}_H(\xi)\boldsymbol{\Omega}^2 &= \begin{bmatrix} \mathbf{R}_\theta & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{E}_B\boldsymbol{\Omega}^2 \\
&= \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi)U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi)U_1 \\ (c_\theta c_\phi)U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}
\end{aligned} \tag{2.48}$$

Dengan menyusun kembali Persamaan (2.43), dapat ditemukan rumus untuk turunan vektor kecepatan yang digeneralisasi terhadap H-frame yang dapat dilihat pada Persamaan (2.49).

$$\dot{\xi} = \mathbf{M}_H^{-1} (\mathbf{C}_H(\xi)\dot{\xi} + \mathbf{G}_H + \mathbf{O}_H(\xi)\boldsymbol{\Omega} + \mathbf{E}_H(\xi)\boldsymbol{\Omega}^2) \tag{2.49}$$

2.1.4 Model Matematika

Dari analisis kinematika dan dinamika diperoleh Persamaan model matematika dari *quadcopter* seperti pada Persamaan (2.50).

$$\begin{cases} \ddot{X} = \frac{U_1}{m} (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \\ \ddot{Y} = \frac{U_1}{m} (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \\ \ddot{Z} = -g + \frac{U_1}{m} (\cos \theta \cos \phi) \\ \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} q r - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} p r - \frac{J_{TP}}{I_{YY}} p \Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} p q - \frac{U_4}{I_{ZZ}} \end{cases} \quad (2.50)$$

Dengan melihat secara sederhana pada Persamaan (2.50), posisi pada sumbu z, dan posisi sudut *roll*, *pitch*, *yaw* dapat dikontrol secara langsung, berturut-turut dengan menggunakan U_1 , U_2 , U_3 , dan U_4 . Kontrol pada posisi maju (x), dan menyamping (y) dapat dilakukan dengan mengatur sudut *pitch* dan *roll* dengan syarat gaya angkat (U_1) tidak sama dengan nol.

Nilai *input* dari *quadcopter* merupakan gaya angkat tiap *propeller*, yang telah dimodelkan secara teoritis dalam Persamaan (2.41), adalah seperti pada Persamaan (2.51).

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = bl(-\Omega_2^2 + \Omega_4^2) \\ U_3 = bl(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (2.51)$$

2.2. Spesifikasi *Plant* Yang Digunakan

Pada subbab ini akan dibahas mengenai spesifikasi dan desain dari *plant quadcopter* yang digunakan.

2.2.1. Sensor *Gyro* dan *Accelerometer* [3]

Sensor *gyro* dan *accelerometer* akan digunakan secara bersamaan untuk menghitung sudut yang berotasi pada sumbu x, y, dan z. Sensor *accelerometer* dan *gyro* yang digunakan adalah MPU6050. Spesifikasinya adalah:

- a) Batas operasi : $\pm 3,6$ g

- b) Sensitivitas : 300 mV/g
- c) Batas suplai tegangan : 1,8 – 5 Volt

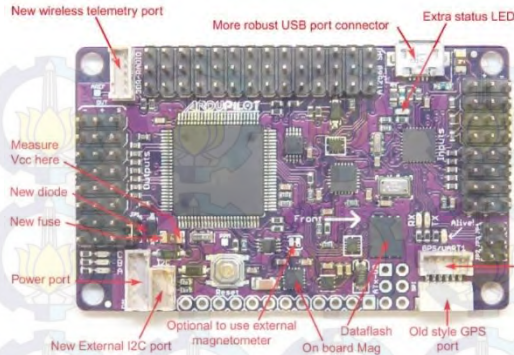
Spesifikasi MEMS *Gyro evaluation board* adalah:

- a) Batas operasi : ± 500 °/s
- b) Sensitivitas : 2,0 mV/°/s
- c) Batas suplai tegangan : 3 Volt.

2.2.2. Ardupilot Mega[4]

Ardupilot Mega merupakan *board* yang mudah digunakan pada *quadcopter*. *Board* ini terdiri atas sensor *gyro* dan *accelerometer*, serta Arduino Mega 2560 sebagai mikrokontrolernya. Arduino Mega berfungsi untuk memproses sinyal dengan program instruksi yang ada. Secara umum APM terdiri atas sambungan I/O, memori internal, dan beberapa moda lain seperti ADC, *Timer/Counter* dan lainnya. Mikrokontroler yang digunakan adalah mikrokontroler ATmega 2560. Mikrokontroler akan diprogram menggunakan bahasa C. Mikrokontroler akan dipadukan menjadi modul Ardupilot Mega dalam satu *board* seperti pada Gambar 2.10. Spesifikasi dari mikrokontroler ATmega 2560 adalah:

- a) Memori : 256KByte
- b) Frekuensi *Clock* : Maksimum 16Mhz
- c) I/O : 86 *Port*
- d) ADC : 10 bit/16 *port*
- e) *Timer* 8 bit : 2 buah
- f) *Timer* 16 Bit : 4 buah
- g) Suplai tegangan : 4,5-5,5 VDC



Gambar 2.10. APM 2.6

2.2.3. Transmitter dan Receiver Radio[5]

Transmitter dan *receiver* digunakan untuk operasi manual melakukan manuver pada *quadcopter*. *Transmitter* berada pada sisi pengguna untuk memberikan nilai *input* kepada *quadcopter*, sinyal tersebut akan diterima *receiver* dan diteruskan ke mikrokontroler berupa pulsa. Tampilan *transmitter* dan *receiver* radio seperti pada Gambar 2.11. Spesifikasi dari radio *transmitter* adalah:

- a) Nama *Transmitter* : FrSky V8FR-II 2.4Ghz 8CH
- b) Channel : 8 buah
- c) Modulasi : 2,4GHz
- d) Baterai : HK 1500mAh Li-Fe
- e) Ukuran : 63,9 x 48,5 x 36,5 mm

Sedangkan, spesifikasi untuk *receiver* adalah:

- a) Nama receiver : FrSky V8FR-II 2.4Ghz 8CH
- b) Ukuran modul utama : 44 x 24 x 14 mm
- c) Berat : 9,3 gram
- d) Range tegangan : 3-16 V
- e) Arus : 30mA



Gambar 2.11. Transmitter dan Receiver

2.2.4. Electronic Speed Controller (ESC) [6]

Alat ini merupakan *driver* dari motor *brushless* DC. ESC di-*trigger* oleh sinyal PWM yang dikendalikan oleh mikrokontroler ATmega 2560. Sinyal PWM tersebut akan men-*drive* motor dengan kecepatan yang linear dengan besar pulsa yang diberikan.

Jenis ESC yang digunakan adalah *TBS Bulletproof 30A* yang dapat dialiri arus hingga 30 Ampere, seperti pada Gambar 2.12. *Quadcopter* membutuhkan respon yang cepat untuk mengatur putaran motor agar lebih stabil. Spesifikasi ESC tersebut adalah:

- | | |
|---------------|----------------------|
| a) Arus | : 30 Ampere |
| b) Tegangan | : 6,4 – 16,8 V |
| c) Lipo Cells | : 2 – 4 buah |
| d) NiMH | : 5 – 12 buah |
| e) Massa | : 32 gram |
| f) Ukuran | : 54x26x11 milimeter |



Gambar 2.12. *Electronic Speed Controller*

2.2.5. Motor BLDC dan Propeller [7]

Motor yang digunakan sebagai aktuator pada implementasi adalah motor DC tanpa sikat *SunnySky X2212 KV980* yang sumbunya dipasang *propeller* berukuran 10x4,5. Bentuk motor DC tanpa sikat dan *propeller* terlihat seperti pada Gambar 3.8. Kombinasi antara motor dan *propeller* tersebut sangat baik untuk mengangkat beban yang berat.



Gambar 2.13. *Motor DC dan Propeller*

2.2.6. Ground Station

Data dari mikrokontroler dikirimkan secara *wireless* menuju komputer *ground station* dengan komunikasi serial menggunakan *telemetry radio 433Mhz*. Data serial langsung dibaca pada *software Mission Planner* pada *ground station*.



Gambar 2.14. Ground Station

2.3. PID Controller [8]

PID *controller* merupakan algoritma metode kendali yang terdiri dari tiga konsep matematika yaitu proporsional, integral, dan derivatif. PID merupakan algoritma kontrol yang paling sering digunakan karena mudah dan memiliki struktur yang sederhana. PID adalah kontroler yang berbasis *negative feedback control*. Pada tugas akhir ini, digunakan kontrol PID untuk mengatur posisi sudut *roll*, *pitch*, dan *yaw* pada *quadcopter*, serta sebagai sinyal kontrol natural pada kontroler *sliding mode*. Persamaan fungsi alih dari kontroler PID ditunjukkan seperti pada Persamaan (2.52).

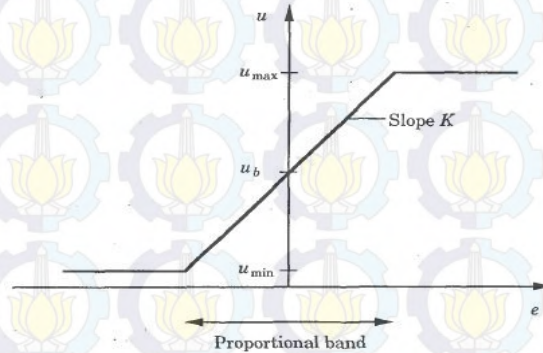
$$u(t) = k_p(e(t)) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \quad (2.52)$$

2.3.1. Proporsional

Kontrol proporsional dikembangkan setelah ketidakpuasan pada kontroler *on-off* yang memberikan dampak osilasi yang sangat besar. Dampak osilasi ini terjadi karena perubahan kecil pada kesalahan akan menghasilkan sinyal kontrol maksimum atau minimum. Karakteristik kontroler proporsional dideskripsikan dengan mengetahui batas sinyal kontrol maksimum dan minimum, serta nilai *band* proporsional yaitu rentang *error*. Maka diketahui bahwa konstanta proporsional dari kontroler adalah *slope* (kemiringan), atau secara matematis dapat dituliskan seperti pada Persamaan (2.53).

$$u_{\max} - u_{\min} = K_{PB} \quad (2.53)$$

Grafik karakteristik dari kontroler ditunjukkan pada Gambar 2.15.



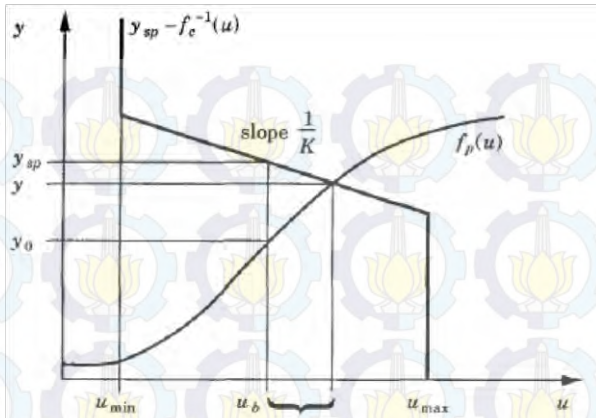
Gambar 2.15 Prinsip Proporsional

Nilai kesalahan keadaan tunak ditunjukkan pada Persamaan (2.54).

$$e = \frac{1}{1 + KK_p} (y_{sp} - y_o) \quad (2.54)$$

Secara grafis karakteristik kontroler proporsional ditampilkan pada Gambar 2.16.

Kontroler proporsional yang semakin besar akan memperkecil kesalahan keadaan tunak, namun akan membawa sistem ke daerah tidak stabil, dan memperkecil rentang operasi nilai *reference*. Sehingga kondisi ini tidak memungkinkan untuk membuat sebuah sistem mempunyai *error steady-state* bernilai 0. Oleh karena itu, dikembangkan kontroler integral untuk menghilangkan *error steady-state*.



Gambar 2.16 Karakteristik Kontroler Proporsional

2.3.2. Integral

Aksi kontrol integral digunakan untuk meyakinkan bahwa respon sistem memiliki nilai kesalahan keadaan tunak sama dengan atau berada di relatif nol. Selama respon sistem memiliki nilai kesalahan, walaupun kecil, secara matematis akan mengakumulasi nilai kesalahan yang mengakibatkan peningkatan nilai sinyal kontrol. Oleh karena itu, aksi kontrol integral akan selalu berusaha untuk menghasilkan respon nilai kesalahan keadaan tunak nol, sehingga mendapatkan nilai aksi kontrol yang konstan. Secara matematis dapat sinyal kontrol keadaan tunak dapat dituliskan sebagai berikut:

$$u(t) = \frac{1}{T_i} \int e(t) dt \quad (2.55)$$

Konsekuensi dengan memperbesar *gain* integral atau memperkecil konstanta waktu integral dapat membuat sistem lebih cepat mengejar nilai keadaan tunak, namun sistem cenderung akan berosilasi. Dalam keadaan seperti ini, diperlukan sebuah metode yang dapat memperkecil amplitudo osilasi dan nilai *overshoot* dari respon.

2.3.3. Derivatif

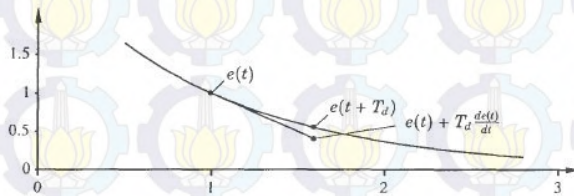
Secara intuitif, respon sistem yang berosilasi disebabkan pada beberapa hal. Proses dinamik dari sebuah plant menyebabkan respon sebuah plant tidak langsung berubah dengan adanya perubahan sinyal

kontrol, namun membutuhkan waktu proses. Waktu ini akan membuat sistem kontrol mengalami keterlambatan untuk mengkoreksi kesalahan. Untuk itu dibutuhkan kontroler yang dapat memprediksi kesalahan dari sebuah sistem seperti pada Gambar 2.17. Nilai kesalahan prediksi dapat diturunkan dari persamaan Taylor seperti pada Persamaan (2.56).

$$e(t + T_d) = e(t) + T_d \frac{de(t)}{dt} \quad (2.56)$$

Dengan mengatur gradien dari garis prediksi, maka akan diperoleh kesalahan prediksi yang lebih akurat.

$$u(t) = k_p (e(t) + T_d \frac{de(t)}{dt}) \quad (2.57)$$



Gambar 2.17 Prediksi Kesalahan

Sebagai validasi, seperti yang telah dipaparkan bahwa osilasi dari respon sistem terjadi karena sinyal kontrol berlebihan yang diberikan ke *plant*. Maka diperlukan mekanisme untuk mengurangi nilai sinyal kontrol yang berlebihan tersebut. Kontroler derivatif yang ditambahkan pada kontroler proporsional dapat mengurangi aksi kontrol yang berlebihan.

Dari Persamaan (2.57), dapat dilihat bahwa keadaan awal saat sistem memiliki kesalahan positif maka kontroler proporsional akan berusaha mengejar nilai *set point*. Dengan adanya respon tersebut, maka perubahan kesalahan proporsional dengan negatif kesalahan. Hal ini akan membuat kontroler derivatif akan berusaha meredam aksi kontroler proporsional yang berlebihan.

2.4. Sliding Mode Control

Pada Tugas Akhir ini digunakan metode *Sliding Mode Control* sebagai algoritma pengaturan posisi *quadcopter* pada sumbu x, y, dan z untuk pesawat *quadcopter*.

2.4.1. Pengenalan *Sliding Mode Control*

Didalam sejarah sistem kontrol, menciptakan dan mendesain *input* yang sesuai sehingga diperoleh *output* yang diinginkan dari sebuah sistem dinamik sudah menjadi perhatian yang utama. Kontroler merupakan sesuatu yang sangat penting dalam menjaga kestabilan sebuah sistem. Karakteristik dari sebuah *plant* atau sistem harus diterima sebagaimana adanya sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu subsistem, yaitu kontroler. Model linear dan strategi kontrol untuk bentuk yang linear sudah dikembangkan dengan cukup baik, namun dalam dunia nyata, kebanyakan sistem-sistem dalam bentuk non linear. Ketidaktepatan permodelan sistem nonlinear bisa disebabkan oleh ketidakpastian kondisi aktual *plant* (parameter *plant* yang berubah-ubah) atau bentuk dinamikanya yang tidak dapat dimodelkan. Contoh-contoh sistem dinamik ialah roket, industri robot, reaktor kimia, dan unit *air conditioner*.

Sliding Mode Control pertama kali dipublikasikan oleh Vadim I. Utkin pada tahun 1977. *Sliding Mode Control* semakin populer beberapa tahun ini, khususnya pada aplikasinya untuk sistem nonlinear dan kemampuannya untuk mempertahankan kekokohnya di banyak bidang. *Sliding Mode Control* termasuk ke dalam kategori *robust control*. *Sliding Mode Controller* mampu menyediakan pendekatan sistematis terhadap masalah mempertahankan stabilitas dan menjaga konsistensi performa dalam bentuk model yang presisi [9]. *Sliding Mode Control* berawal dari kontrol *on-off* yang semakin lama semakin berkembang. Istilah *sliding mode* pertama kali muncul didalam konteks sistem relay dan dalam merupakan pengembangan dari teori *Bang-Bang control*. Sinyal kontrol sebagai fungsi dari sistem mengalami *switch* (perubahan) dengan frekuensi tinggi.

Sliding Mode Control merupakan salah satu metode kontrol untuk sistem linear dan nonlinear. Metode kontrol yang dilakukan yaitu dengan memaksa trayektori *state* suatu sistem untuk menuju ke dalam sebuah permukaan luncur tertentu. Pergerakan ini biasa disebut dengan *sliding mode*. Setelah sampai pada permukaan luncur, sinyal kontrol akan berusaha mempertahankan trayektori *state* tetap berada sepanjang permukaan luncur [10]. *Sliding Mode Control* telah berhasil diaplikasikan pada *robot manipulator*, kapal selam, mesin dan penyebaran otomotif, motor elektrik *high-performance* dan sistem tenaga [11].

Metode *Sliding Mode Control* memiliki sifat tidak sensitif terhadap perubahan parameter *plant* dan gangguan eksternal. *Sliding*

Mode Control merupakan suatu metode kontrol *state feedback* yang menggunakan kontrol penyaklaran berkecepatan tinggi. Hal ini mengakibatkan trayektori *state* disekitar permukaan luncur berosilasi dengan frekuensi tinggi atau yang biasa disebut dengan fenomena *chattering*. Osilasi tersebut mempengaruhi stabilitas pada sistem.

Salah satu aspek yang menarik dari *sliding mode* adalah sifat diskontinyu pada aksi kontrolnya yang memiliki fungsi utama pada setiap saluran *feedback* untuk melakukan *switch* pada *sliding surface* (*manifold*) yang ada [9]. *Sliding mode* dapat digunakan tidak hanya pada *error* dan turunannya, tetapi pada semua variabel *state* [10]. Kelebihan dari *Sliding Mode Control* adalah kekurang-pekaannya terhadap kesalahan model, ketidakpastian parameter, dan gangguan lainnya [11].

Sliding Mode Control bekerja untuk memaksa *state* sistem untuk menuju permukaan luncur (*sliding surface*) dan mempertahankan *state* tersebut untuk tetap berada pada permukaan luncurnya. Pada subbab ini akan dibahas mengenai analisis kestabilan Lyapunov, permukaan luncur, dan sinyal pengaturan pada *Sliding Mode Control*.

2.4.2. Analisis Kestabilan Lyapunov [10], [12]

Pada sistem pengaturan, kestabilan merupakan hal yang paling utama. Metode Lyapunov yang kedua adalah metode yang paling umum digunakan untuk menentukan stabilitas suatu sistem linear maupun nonlinear. Dengan metode ini, stabilitas suatu sistem dapat ditentukan tanpa memecahkan persamaan *state* sistem tersebut, sehingga metode ini sangat bermanfaat karena pemecahan persamaan *state* untuk sistem nonlinear sangat sulit. Lyapunov mengatakan bahwa apabila suatu sistem memiliki sebuah ekuilibrium *state* yang stabil \mathbf{x}_e , maka total energi yang tersimpan pada sistem akan berangsur-angsur berkurang seiring dengan bertambahnya waktu (t), hingga total energi mencapai nilai minimumnya di ekuilibrium *state* \mathbf{x}_e . Fungsi Lyapunov dapat dibentuk dari vektor *state* sistem \mathbf{x} yang kemudian dinyatakan dalam fungsi Lyapunov $V(\mathbf{x})$. Informasi mengenai kestabilan sistem dapat diperoleh hanya dengan mengamati fungsi $V(\mathbf{x})$ serta turunan parsial pertamanya $dV(\mathbf{x})/dt$ tanpa harus memperoleh solusi persamaan diferensial sistem. $V(\mathbf{x})$ harus memenuhi kondisi berikut untuk semua nilai $t_1 > t_0$ dan untuk semua \mathbf{x} di sekitar titik ekuilibrium $\mathbf{x} = 0$:

- a) $V(\mathbf{x})$ dan turunan parsialnya dapat didefinisikan dan kontinyu
- b) $V(\mathbf{0}) = 0$
- c) $V(\mathbf{x}) > 0$, untuk semua nilai $\mathbf{x} \neq 0$

d) $\dot{V}(\mathbf{x}) < 0$, untuk semua nilai $\mathbf{x} \neq \mathbf{0}$

Metode kedua Lyapunov didasarkan pada teorema berikut:

Teorema 1

Misalkan ada suatu sistem

$$\dot{\mathbf{x}} = f(\mathbf{x}, t), f(\mathbf{0}, t) = \mathbf{0} \quad (2.58)$$

Diasumsikan bahwa $V(\mathbf{x})$ pada Persamaan (2.58) dapat ditentukan. Oleh karena itu, ekuilibrium *state* $\mathbf{x} = \mathbf{0}$ dapat dikatakan stabil asimtotik dan sistem pada Persamaan (2.58) dikatakan *stable in the sense of Lyapunov*.

Teorema 2

Misalkan ada suatu sistem linear *time invariant* $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, dengan $|\mathbf{A}| \neq 0$ dan $\mathbf{x}_e = \mathbf{0}$. Dan misalkan fungsi skalar $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$, dengan \mathbf{P} merupakan matriks simetris yang juga definit positif. $V(\mathbf{x})$ merupakan fungsi Lyapunov jika dan hanya jika untuk setiap matriks simetris definit positif \mathbf{Q} dapat ditentukan matriks simetris definit positif \mathbf{P} yang memenuhi Persamaan (2.59).

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q} \quad (2.59)$$

Pembuktian dari Persamaan (2.59) adalah sebagai berikut. Turunan dari fungsi Lyapunov $V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$ ditunjukkan pada Persamaan (2.59). Persamaan untuk \mathbf{x} merupakan persamaan *state* untuk sistem *autonomous*, seperti pada Persamaan (2.60), sehingga turunan fungsi Lyapunov $\dot{V}(\mathbf{x})$ adalah Persamaan (2.61).

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} \quad (2.60)$$

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} \quad (2.61)$$

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T (\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A}) \mathbf{x} \quad (2.62)$$

Berdasarkan definisi yang diberikan pada awal subbab ini, dapat dilihat kondisi d, yaitu $\dot{V}(\mathbf{x}) < 0$ untuk semua $\mathbf{x} \neq \mathbf{0}$. Turunan fungsi Lyapunov dapat dibuat dalam Persamaan (2.63).

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T(\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A})\dot{\mathbf{x}} = -\mathbf{x}^T\mathbf{Q}\mathbf{x} \quad (2.63)$$

Ruas sebelah kanan dapat dituliskan $-\mathbf{x}^T\mathbf{Q}\mathbf{x} < 0$ dengan pemilihan nilai \mathbf{Q} . Dari Persamaan (2.62), matriks \mathbf{P} dan \mathbf{A} harus memenuhi Persamaan (2.63).

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q} \quad (2.64)$$

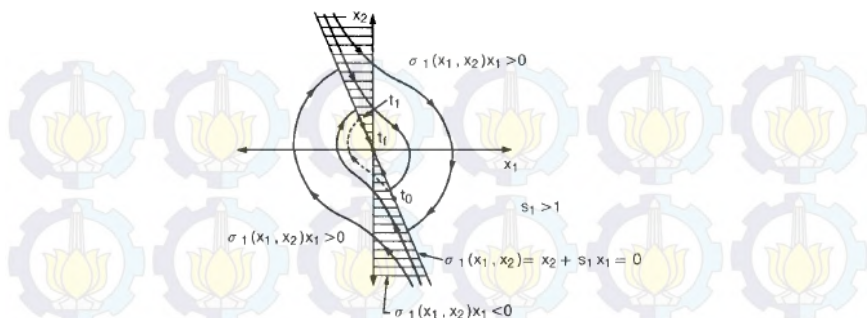
2.4.3. Permukaan Luncur

Semua *state-space* dari suatu sistem yang menyebabkan kondisi *sliding mode* dapat terjadi disebut *sliding manifold*, atau *sliding surface* (permukaan luncur). Perilaku dinamis dari suatu sistem saat selalu menuju *surface* tertentu disebut *sliding motion* yang ideal [9]. Sebuah sistem yang mencapai sebuah *motion* (gerakan) memiliki keuntungan dua kali lipat: pertama terjadi pereduksian orde sistem dan kedua *sliding surface* tidak sensitif terhadap parameter yang mutlak pada saluran inputnya [13].

Pemilihan permukaan luncur dilakukan dengan pertimbangan agar *state* sistem dapat menuju permukaan luncur dari sembarang *initial condition* dan agar *state* sistem dapat dipertahankan di sekitar permukaan luncur. *Switching* dapat terjadi dengan adanya permukaan luncur, yang secara umum dinotasikan $\sigma = 0$ [11]. Ide ini dapat diilustrasikan dengan memisalkan permukaan luncur ditentukan sebagai $\sigma = \sigma_1(x_1, x_2) = s_1x_1 + x_2 = 0$ dengan $s_1 > 1$. Jika *feedback* sistem di-switch menurut

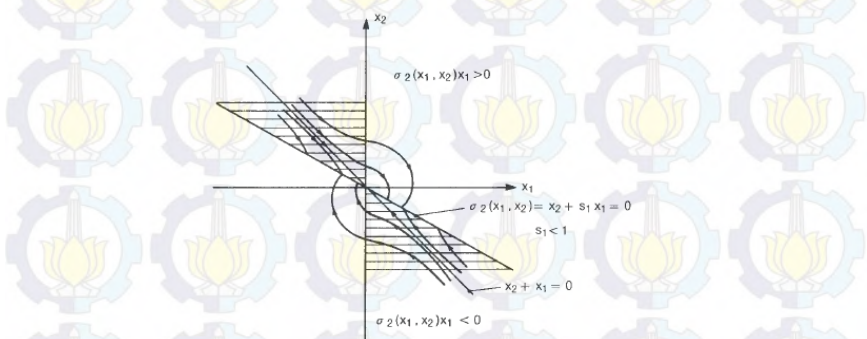
$$k(x_1) = \begin{cases} -3, & \text{jika } \sigma_1(x_1, x_2)x_1 > 0 \\ 2, & \text{jika } \sigma_1(x_1, x_2)x_1 < 0 \end{cases} \quad (2.65)$$

Perilaku yang diilustrasikan dalam Gambar 2.20 menunjukkan hasil plot *phase plane* [11].



Gambar 2.18 Jalur (Garis yang Diberi Titik) dari Lintasan *State* dari Suatu Sistem dengan Kontroler Saat Diberi Gangguan Kecil [11]

Dengan mengamati garis yang diberi titik dalam lintasan *state* ketika mengalami gangguan kecil, $\sigma_1(x_1, x_2) = s_1x_1 + x_2 = 0$, pada saat t_0 , melingkar di titik t_1 sebelum kembali ke permukaan luncur lagi. Disisi lain, jika permukaan *switching* adalah $\sigma_1(x_1, x_2) = s_1x_1 + x_2 = 0$ dengan $s_1 < 1$, maka gangguan yang terjadi akan dipaksa kembali pada permukaan luncur karena vektor kecepatan *phase-plane* selalu menunjuk pada permukaan. Gambar 2.21 menggambarkan fenomena ini.



Gambar 2.19 Penggambaran *Phase-Plane* dari Sistem yang Dikontrol dengan Menggunakan SMC pada *Switching Surface*

Kedua gambar ini menunjukkan bahwa pilihan *switching surface* yang berbeda mengakibatkan bentuk respon sistem yang sangat berbeda. Mendesain permukaan luncur merupakan hal yang penting dalam mendesain *Sliding Mode Controller*.

2.4.4. Sinyal Kontrol

Tujuan dari perancangan *Sliding Mode Control* adalah memaksa trayektori *state* menuju permukaan luncur dan menjaga trayektori *state* agar tetap berada di sekitar permukaan luncur. Ada dua jenis sinyal kontrol yang akan dirancang, yaitu sinyal kontrol ekuivalen U_{eq} dan sinyal kontrol natural U_N , di mana sinyal kontrol keseluruhan adalah penjumlahan dari keduanya.

$$U = U_{eq} + U_N \quad (2.66)$$

Metode kontrol ekuivalen dapat digunakan untuk menetapkan pergerakan pada batas permukaan luncur $\sigma(x) = 0$. *Sliding mode* terjadi pada saat $\sigma(x) = 0$, untuk semua $t \geq t_0$ dan $\dot{\sigma}(x) = 0$.

Sinyal kontrol ekuivalen U_{eq} merupakan kontroler yang memaksa trayektori sistem agar menuju ke permukaan luncur, sedangkan sinyal kontroler natural U_N adalah sinyal kontrol yang menjaga trayektori sistem agar tetap berada di sekitar permukaan luncur.

2.4.5. *Sliding Mode Control* Berbasis PID [14],[15],[16]

Sliding Mode Control Berbasis PID merupakan salah satu pengembangan dari *Sliding Mode Control* konvensional. PID-SMC menggunakan sinyal kontrol natural berupa struktur PID, seperti pada Persamaan (2.67).

$$U_{PID} = k_p s + k_i \int s dt + k_d \frac{ds}{dt} \quad (2.67)$$

Sinyal kontrol natural yang menyerupai struktur PID ini diharapkan dapat menghilangkan *chattering* yang ditimbulkan oleh fungsi diskontinyu. Aksi dari sinyal kontrol integral akan membuat *state* lebih cepat menuju *sliding surface* yang direncanakan. Sedangkan, aksi kontrol

dari sisi derivatif akan menstabilkan sistem dengan melawan aksi kontrol berlebihan yang dapat ditimbulkan oleh integral. [14].

Pemilihan nilai k_p , k_i , dan k_d dilakukan sedemikian rupa sehingga memenuhi syarat kestabilan Lyapunov dari *Sliding Mode Control*.

2.5. Linearisasi [17]

Pada suatu sistem, seringkali hubungan antara besaran-besaran dalam sistem tersebut adalah tidak linear. *Quadcopter* adalah salah satu sistem yang mempunyai hubungan-hubungan antar variabel yang tidak linear. Hal ini terlihat pada Persamaan (2.50), dimana antara satu variabel dengan variabel lain mempunyai hubungan berupa trigonometri, yang merupakan hubungan tidak linear.

Untuk dapat melakukan proses kontrol yang menggunakan kontroler linear pada *plant* yang nonlinear, maka sistem atau *plant* tersebut harus dilinearisasi terlebih dahulu. Teori linearisasi digunakan untuk mengatasi permasalahan ketidaklinearan sistem, sehingga sistem yang nonlinear dapat didekati dengan sistem yang linear. Pada penelitian ini, digunakan teknik linearisasi *feedback linearization*. *Feedback linearization* merupakan suatu teknik linearisasi dengan melakukan modifikasi pada variabel yang dikontrol sehingga sistem dapat didekati dengan sistem linear.

Feedback linearization digambarkan sebagai berikut. Misalkan suatu sistem nonlinear mempunyai persamaan *state* seperti pada Persamaan (2.67).

$$\begin{aligned}\dot{x} &= f(x) + G(x)u \\ y &= h(x)\end{aligned}\tag{2.67}$$

Dimana $f: D \rightarrow \mathbb{R}^n$ dan $G: D \rightarrow \mathbb{R}^{n \times m}$ didefinisikan pada domain $D \subset \mathbb{R}^n$, dimana domain tersebut mengandung origin. Sehingga akan didapatkan sinyal kontrol modifikasi seperti pada Persamaan (2.68).

$$u = \alpha(x) + \beta(x)v\tag{2.68}$$

BAB III

PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan-perancangan sistem dari *quadcopter* yang terdiri atas spesifikasi, identifikasi kebutuhan, komponen mekanik dan elektronik, model matematik, identifikasi konstanta, perancangan kontroler, dan simulasi dari sistem yang akan dibuat.

3.1. Spesifikasi Sistem

Quadcopter sebagai pesawat tanpa awak yang dapat bergerak dengan 6 DOF (*degree of freedom*), terdiri dari 3 DOF rotasi dan 3 DOF translasi, akan dibuat menjadi spesifikasi sistem tertentu secara *hardware* maupun simulasi pada *software* MATLAB dapat diuraikan sebagai berikut:

- a) *Quadcopter* dapat bergerak 6 derajat kebebasan, dan dapat mengikuti gerak translasi pada sumbu x maupun sumbu y.
- b) *Quadcopter* dapat digerakkan secara manual dengan menggunakan *remote control*.
- c) Data-data sensor dan aktuator dikirimkan ke *ground station*.

3.2. Identifikasi Kebutuhan

Untuk memenuhi spesifikasi sistem yang diharapkan maka terdapat kebutuhan yang harus dipenuhi antara lain:

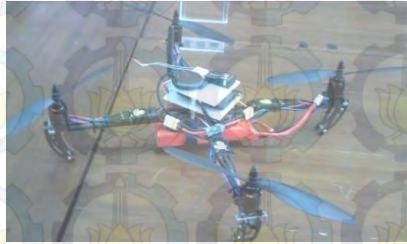
- a) *Quadcopter* harus dibuat simetris dan ringan untuk menghindari kelembaman yang sangat besar.
- b) Baterai yang digunakan minimal dapat membuat *quadcopter* terbang sekitar 15 menit.
- c) Rangkaian mikrokontroler memiliki memori dan *port I/O* yang cukup untuk sensor, aktuator dan komunikasi.
- d) Data-data selama terbang dapat dikirimkan ke *ground station*.

3.3. Desain Mekanik dan Elektronik Sistem

Subbab ini membahas desain mekanik dan desain elektronik dari sistem yang digunakan.

3.3.1. Desain Mekanik

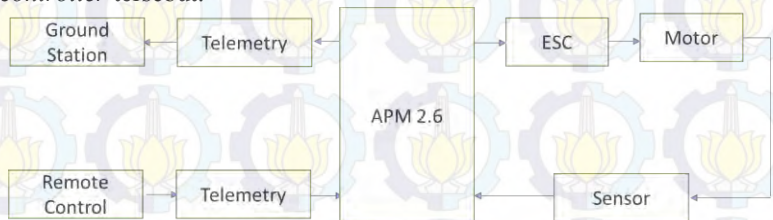
Desain mekanik dari *quadcopter* harus dibuat simetris dan cukup ringan. Frame *quadcopter* yang digunakan adalah berbentuk *plus (+)*. Desain mekanik yang dibangun untuk implementasi mode *waypoint tracking* pada penelitian Tugas Akhir ini ditunjukkan pada Gambar 3.1.



Gambar 3.1. Desain Mekanik *Quadcopter* yang Dibuat

3.3.2. Desain Elektronik

Board kontroler yang dipakai adalah Ardupilot Mega 2.6 yang sudah dilengkapi beberapa sensor dan *port* yang memudahkan untuk penggunaan sebagai komunikasi data. *Board* Ardupilot Mega 2.6 ini dilengkapi dengan ATmega 2560 sebagai mikrokontroler dari *flight controller* tersebut.



Gambar 3.2. Rancangan Sistem Elektronika *Quadcopter*

3.4. Pemodelan Motor-Propeller

Untuk menemukan model linear dari motor, maka dibutuhkan eksperimen. Terdapat 2 buah eksperimen yang dilakukan, yaitu eksperimen untuk mengetahui model linear dari hubungan pulsa dengan kecepatan motor (rpm dan rad/s), dan eksperimen untuk mengetahui hubungan antara pulsa dengan gaya angkat motor (dalam kg).

3.4.1. Pengukuran Kecepatan Putar Motor

Pada eksperimen 1 ini dilakukan pengukuran kecepatan motor dengan menggunakan tachometer, dengan sinyal masukan berupa sinyal PWM dari Arduino. Hasil pengukuran kecepatan motor ditampilkan pada Lampiran A.

Dari hasil pengukuran diperoleh persamaan linear hubungan pulsa motor terhadap kecepatan (rad/s) pada masing-masing motor adalah sebagai berikut :

$$\begin{aligned}y_1 &= 44,851x - 1760,5 \\y_2 &= 44,10x - 1603,7 \\y_3 &= 44,8x - 1670,4 \\y_4 &= 45,047x - 1618,7\end{aligned}\tag{3.1}$$

3.4.2. Pengukuran Gaya Angkat *Propeller*

Langkah pengukuran gaya angkat *propeller* yaitu dengan meletakkan motor dan *propeller* tepat diatas beban 1,2 kg, yaitu botol yang terisi air, kemudian motor dijalankan dengan memberikan pulsa atau sinyal PWM dari mikrokontroler. Hasil pembacaan berat pada timbangan dikurangi dengan beban 1,2 kg adalah gaya angkat pada masing-masing motor. Pada eksperimen ini dilakukan pengukuran gaya angkat pada masing-masing motor *propeller*.

Dari pengukuran diperoleh persamaan linear hubungan pulsa motor terhadap gaya angkat (kilogram) pada masing-masing motor adalah seperti pada Persamaan (3.2).

$$\begin{aligned}y_1 &= 0,0076x - 0,6097 \\y_2 &= 0,0076x - 0,5968 \\y_3 &= 0,0076x - 0,5968 \\y_4 &= 0,0078x - 0,6026\end{aligned}\tag{3.2}$$

3.4.3. Pengukuran Hubungan *Throttle* Pada *Remote Control* dan Kecepatan Motor

Pengukuran ini bertujuan untuk mengetahui hubungan antara nilai persentase *throttle* pada *remote control* dan kecepatan motor. Percobaan yang dilakukan adalah dengan memberikuan input yang berbeda-beda pada *remote control*, lalu kecepatan tiap-tiap motor diukur dengan menggunakan tachometer. Dari percobaan yang telah dilakukan,

didapatkan hasil persamaan linear hubungan *throttle* dan kecepatan motor seperti pada Persamaan (3.3).

$$\begin{aligned}y_1 &= 4,6944x + 2646,8 \\y_2 &= 25,018x + 3788,6 \\y_3 &= 37,712x + 486,86 \\y_4 &= 38,088x + 3191,5\end{aligned}\tag{3.3}$$

Dimana y_n adalah kecepatan masing-masing motor dalam rpm ($n=1,2,3,4$), dan x adalah presentase *throttle* pada *remote control*.

3.5. Identifikasi dan Validasi Sistem

Dari model matematik yang diperoleh dengan pemodelan fisis pada Bab II, maka terlihat bahwa dibutuhkan konstanta-konstanta yang terdapat pada sistem agar model matematika bisa digunakan pada simulasi dan penentuan parameter-parameter kontroler. Beberapa konstanta yang harus diperoleh antara lain:

- a) Massa
- b) Panjang lengan *quadcopter*
- c) Momen inersia
- d) Pemetaan gaya angkat terhadap pulsa
- e) Konstanta *thrust*
- f) Konstanta *drag*

Massa dari sistem keseluruhan diukur menggunakan alat pengukur massa. Dari hasil pengukuran diperoleh massa total sistem dan panjang lengan *quadcopter*. Massa total dari *quadcopter* adalah 1,26 kg. Sedangkan, panjang lengan *quadcopter* adalah 0,206 m.

Momen inersia dari sistem terhadap sumbu x , sumbu y , dan sumbu z diperoleh dengan identifikasi parametrik dari *input-output* sistem dengan cara menerbangkan *quadcopter* lalu didapatkan data keluaran dari sensor yaitu berupa kecepatan sudut, serta nilai persentase *throttle* dari *remote control*.

Sedangkan, untuk mendapatkan nilai konstanta *thrust* dan konstanta *drag*, dilakukan pengujian pada masing-masing motor dengan diberi suplai sinyal PWM, lalu diukur kecepatan dari keempat motor tersebut, dan dihitung konstanta *thrust* dan *drag* dari *quadcopter*.

Untuk mendapatkan konstanta *thrust*, *quadcopter* diterbangkan dalam posisi *hover*, karena pada posisi *hover*, gaya yang dihasilkan oleh motor untuk mengangkat *quadcopter* sama dengan gaya ke bawah.

Dari percobaan yang telah dilakukan, dapat diketahui kecepatan tiap-tiap motor pada saat keadaan *hover*. *Quadcopter* yang digunakan memiliki massa total sebesar 1,26 kg dan percepatan gravitasi adalah $9,81 \text{ m/s}^2$. Dengan memasukkan data-data tersebut ke persamaan maka diperoleh konstanta *thrust* sebesar $1,6798 \times 10^{-5}$.

Konstanta *drag* dihitung dengan persamaan gerak lurus berubah beraturan. Pengukuran konstanta *drag* dilakukan dengan mengambil data penerbangan *quadcopter* pada saat *take-off*. Prosedur penghitungan konstanta *thrust* dan *drag* ditampilkan dalam Lampiran B.

Hasil identifikasi konstanta *drag*, *thrust*, massa, dan jari-jari pada *quadcopter* ditunjukkan pada Tabel 3.1.

Tabel 3.1 Parameter *Quadcopter*

Parameter	Nilai	Satuan
Massa <i>quadcopter</i> (m)	1,26	kg
Jari-jari <i>quadcopter</i> (l)	0,206	Meter
Konstanta <i>Thrust</i>	$1,68918 \times 10^{-5}$	N.sec ²
Konstanta <i>Drag</i>	$4,19 \times 10^{-6}$	Nm.sec ²

Model matematika yang digunakan untuk identifikasi adalah model matematika rotasi, karena langsung berhubungan dengan konstanta-konstanta yang belum diketahui pada *quadcopter*. Untuk menyederhanakan perhitungan, maka model matematika rotasi dari *quadcopter* dimisalkan seperti pada Persamaan (3.4).

$$\begin{aligned}
 \dot{p} &= a_1 q r + b_1 q \Omega + c_1 U_2 \\
 \dot{q} &= a_2 p r + b_2 p \Omega + c_2 U_3 \\
 \dot{r} &= a_3 p q + c_3 U_4
 \end{aligned} \tag{3.4}$$

Dimana Persamaan (3.4) diturunkan dari Persamaan (2.50). Langkah-langkah identifikasi secara parametrik adalah sebagai berikut :

- Quadcopter* dioperasikan, dengan menggunakan nilai *throttle* yang bervariasi pada remote control,
- Data yang dari sensor (kecepatan sudut, posisi sudut, persentase nilai *throttle*) direkam melalui *ground station*,
- Data tersebut kemudian dianalisis dengan menggunakan metode penyelesaian persamaan simultan,,
- Dari proses identifikasi tersebut didapatkan konstanta-konstanta yang dibutuhkan,

- e) Dilakukan proses validasi dengan cara membandingkan dengan identifikasi melalui perhitungan fisis.

Sedangkan, identifikasi secara fisik dilakukan dengan menghitung secara fisik besaran-besaran yang tidak diketahui, seperti momen inersia dan konstanta motor-*propeller*. Perhitungan identifikasi secara fisik dijelaskan dalam Lampiran C.

Melalui identifikasi yang dilakukan, didapatkan parameter *quadcopter* sebagai berikut :

Tabel 3.2 Konstanta Hasil Identifikasi

Konstanta	Hasil Identifikasi Parametrik	Hasil Identifikasi Fisik
a_1	-0,5495	0,25327
b_1	-0,0017	-0,0198
c_1	0,2052	595,174
a_2	0,1675	-0,2537
b_2	-0,0094	0,0198
c_2	2,955	595,174
a_3	-2,0257	0
c_3	0,0594	797,04

Setelah didapatkan konstanta, selanjutnya akan dilakukan proses validasi yaitu membandingkan konstanta yang dihasilkan dengan persamaan sistem yang sesungguhnya. Konstanta yang didapat kemudian dimasukkan kembali ke Persamaan (3.4) dan dibandingkan dengan nilai pembacaan sensor yang sesungguhnya.

Tabel hasil proses validasi dengan menggunakan *RMS error* sebagai pembanding ditunjukkan pada Tabel 3.3.

Tabel 3.3 Perbandingan Hasil Identifikasi

Besaran Yang Dihitung	RMSE Identifikasi Parametrik	RMSE Identifikasi Fisik
\dot{p}	3,2232%	190,34%
\dot{q}	1,3507%	177,53%
\dot{r}	5,3405%	92,525%

Dari proses validasi tersebut, didapatkan model matematika yang cocok adalah model matematika pada identifikasi secara parametrik.

3.6. Model Matematika Hasil Identifikasi Sistem

Setelah diperoleh konstanta-konstanta dari sistem, maka dapat dituliskan kembali model matematika dari sistem *quadcopter* adalah sebagai berikut:

$$\begin{aligned}
 \ddot{X} &= \frac{U_1}{1,26} (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \\
 \ddot{Y} &= \frac{U_1}{1,26} (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \\
 \ddot{Z} &= \frac{U_1}{1,26} (\cos \theta \cos \phi) - 9,81 \\
 \dot{p} &= -0,5495qr - 0,00017q\Omega + 0,2052U_2 \\
 \dot{q} &= 0,1675pr - 0,0094p\Omega + 2,955U_3 \\
 \dot{r} &= -2,0257pq + 0,0594U_4
 \end{aligned} \tag{3.5}$$

Dari identifikasi konstanta dilakukan pembatasan, di antaranya sudut *roll* dan *pitch* adalah maksimum 0,7 rad, dan gaya angkat maksimum U_1 adalah 13 N.

3.7. Perancangan Kontroler

Setelah diperoleh model matematika dan konstanta maka dapat dilakukan perancangan kontroler dengan mencari parameter-parameter kontroler yang diperlukan.

Pada perancangan *Sliding Mode Control* berbasis PID, diperlukan prosedur untuk memperoleh struktur dan parameter kontroler. Prosedur

perancangan *Sliding Mode Control* dilakukan melalui beberapa tahap, yaitu: perancangan permukaan luncur, perancangan kontroler ekuivalen, perancangan kontroler natural, dan sinyal kontrol total. Tujuan pengaturan atau *objective control* pada Tugas Akhir ini adalah *quadcopter* dapat melakukan gerakan *take-off*, *hover*, dan *landing*, serta dapat melakukan *tracking* sesuai dengan referensi yang diinginkan.

Berdasarkan model yang telah diperoleh pada Bab II, akan dilakukan perancangan *PID-Based Sliding Mode Control* sebagai kontroler posisi x , y , dan z , dan dilakukan perancangan kontroler *cascade* P dan PD sebagai kontroler sudut *roll*, *pitch*, dan *yaw*.

Untuk mempermudah perhitungan, model *quadcopter* dapat disusun dalam bentuk *state space*:

$$\dot{X} = f(X) + g(X, U) + \delta; \quad X = [x_1 \dots x_{12}]^T \quad (3.6)$$

Dipilih masing-masing *state* pada Persamaan (3.6) adalah sebagai berikut :

$$\begin{aligned} x_1 &= x \\ x_2 &= \dot{x} \\ x_3 &= y \\ x_4 &= \dot{y} \\ x_5 &= z \\ x_6 &= \dot{z} \\ x_7 &= \phi \\ x_8 &= \dot{\phi} = p \\ x_9 &= \theta \\ x_{10} &= \dot{\theta} = q \\ x_{11} &= \psi \\ x_{12} &= \dot{\psi} = r \end{aligned} \quad (3.7)$$

Representasi *state* pada Persamaan (3.8) dapat diperoleh dari persamaan model *quadcopter* dan Persamaan (3.7).

$$\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = \frac{U_1}{m} (\sin x_{11} \sin x_7 + \cos x_{11} \sin x_9 \cos x_7) \\
\dot{x}_3 = x_4 \\
\dot{x}_4 = \frac{U_1}{m} (-\cos x_{11} \sin x_7 + \sin x_{11} \sin x_9 \cos x_7) \\
\dot{x}_5 = x_6 \\
\dot{x}_6 = -g + \frac{U_1}{m} (\cos x_9 \cos x_7) \\
\dot{x}_7 = x_8 \\
\dot{x}_8 = \frac{I_{YY} - I_{ZZ}}{I_{XX}} x_{10} x_{12} - \frac{J_{TP}}{I_{XX}} x_{10} \Omega + \frac{U_2}{I_{XX}} \\
\dot{x}_9 = x_{10} \\
\dot{x}_{10} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} x_8 x_{12} - \frac{J_{TP}}{I_{YY}} x_8 \Omega + \frac{U_3}{I_{YY}} \\
\dot{x}_{11} = x_{12} \\
\dot{x}_{12} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} x_8 x_{10} - \frac{U_4}{I_{ZZ}}
\end{cases} \quad (3.8)$$

Maka Persamaan (3.8) dapat ditulis seperti pada Persamaan (3.9).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_5 \\ \frac{U_1}{m} U_x \\ x_4 \\ \frac{U_1}{m} U_y \\ x_6 \\ -g + \frac{U_1}{m} (\cos x_9 \cos x_7) \\ x_8 \\ \frac{I_{YY} - I_{ZZ}}{I_{XX}} x_{10} x_{12} - \frac{J_{TP}}{I_{XX}} x_{10} \Omega + \frac{U_2}{I_{XX}} \\ x_{10} \\ \frac{I_{ZZ} - I_{XX}}{I_{YY}} x_8 x_{12} - \frac{J_{TP}}{I_{YY}} x_8 \Omega + \frac{U_3}{I_{YY}} \\ x_{12} \\ \frac{I_{XX} - I_{YY}}{I_{ZZ}} x_8 x_{10} - \frac{U_4}{I_{ZZ}} \end{bmatrix} \quad (3.9)$$

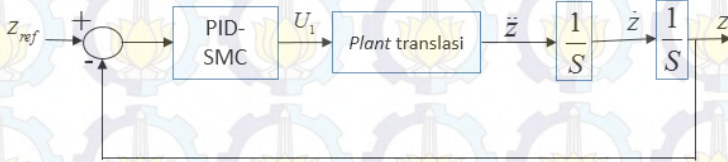
Dimana

$$U_x = (\sin x_{11} \sin x_7 + \cos x_{11} \sin x_9 \cos x_7) \quad (3.10)$$

$$U_y = (-\cos x_{11} \sin x_7 + \sin x_{11} \sin x_9 \cos x_7) \quad (3.11)$$

3.7.1. Pengaturan Posisi Sumbu Z

Ketinggian pada *quadcopter* berhubungan dengan sumbu z . *State* \dot{x}_5 dan \dot{x}_6 yang merepresentasikan persamaan untuk ketinggian *quadcopter* dari bumi, diambil dari model *quadcopter* pada Persamaan (3.4). Skema sistem pengaturan ketinggian untuk gerak *vertical take-off* dan *landing* pada *quadcopter* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Blok Pengaturan Ketinggian *Quadcopter*

Persamaan *state quadcopter* untuk ketinggian dapat dituliskan sebagai berikut:

$$\begin{cases} \dot{x}_5 = x_6 \\ \dot{x}_6 = -g + (\cos x_9 \cos x_7) \frac{U_1}{m} \end{cases} \quad (3.12)$$

Persamaan (3.12) dapat dibuat dalam bentuk *state space* seperti Persamaan (3.13).

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_6 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{(\cos x_9 \cos x_7)}{m} \end{bmatrix} U_1 \quad (3.13)$$

Persamaan (3.13) dapat dituliskan dengan persamaan *state* seperti pada Persamaan (3.14).

$$\dot{z} = f(X) + g(x)U_1 \quad (3.14)$$

Langkah selanjutnya adalah mendesain permukaan luncur di mana *Sliding Mode* akan terjadi. Sebelum mendesain permukaan luncur, perlu ditentukan terlebih dahulu trayektori *state* yang akan dijadikan permukaan luncur, dalam hal ini berupa fungsi dari *error* sistem.

Respon sistem yang diinginkan adalah orde pertama. Persamaan (3.15) merupakan persamaan untuk respon sistem orde pertama.

$$Y_c(s) = \frac{1}{\tau s + 1} Y_r(s) \quad (3.15)$$

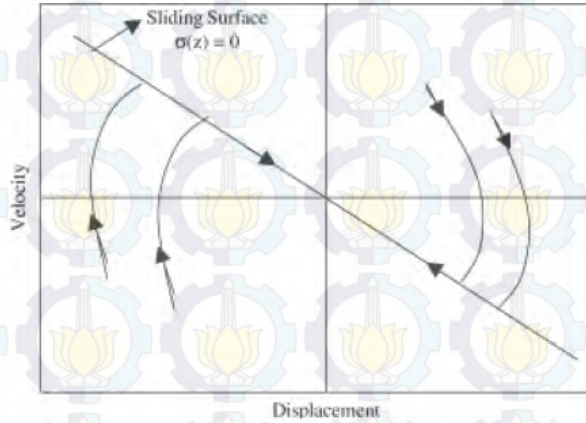
Untuk mendapatkan sinyal kontrol ekuivalen (U_{eq}), maka didefinisikan *error tracking* yang merupakan perbedaan antara nilai posisi *quadcopter* dengan posisi referensi sebagai berikut :

$$e_s = x_{sref} - x_s \quad (3.16)$$

Jika referensi (s) merupakan *step* dan turunannya sama dengan nol ($\dot{Y}(s) = 0$), maka akan diperoleh persamaan *error* seperti pada Persamaan (3.17).

$$\tau e_s + \dot{e}_s = 0 \quad (3.17)$$

Permukaan luncur yang akan digunakan adalah persamaan *error*, yang dapat digambarkan dalam bidang fasa, seperti pada Gambar 3.4.



Gambar 3.4 Permukaan Luncur

Dengan x_{sref} adalah sinyal referensi. Selanjutnya akan dirancang permukaan luncur (*sliding surface*), yang didefinisikan sebagai berikut :

$$s_s = \tau e_s + \dot{e}_s \quad (3.18)$$

Untuk membuat permukaan luncur menuju 0 pada waktu tak hingga, maka s diturunkan terhadap waktu, sehingga didapatkan

$$\dot{s}_s - \tau \dot{e}_s + \ddot{e}_s = 0 \quad (3.19)$$

Substitusi e_s dengan Persamaan (3.16), maka didapatkan

$$\dot{s}_s = \tau \dot{e}_s - \ddot{x}_s + \ddot{x}_{sref} = 0 \quad (3.20)$$

Dengan $\ddot{x}_s = \dot{x}_6$, maka didapatkan

$$\dot{s}_s = \tau \dot{e}_s - \dot{x}_6 + \ddot{x}_{sref} = 0 \quad (3.21)$$

Substitusi dengan Persamaan (3.9), didapatkan

$$\dot{s}_5 = \tau \dot{e}_5 + \ddot{x}_{5ref} - \frac{U_1}{m} \cos x_7 \cos x_9 + g = 0 \quad (3.22)$$

$$\tau \dot{e}_5 + \ddot{x}_{5ref} + g = \frac{U_1}{m} \cos x_7 \cos x_9 \quad (3.23)$$

$$U_1 = \frac{m(\tau \dot{e}_5 + \ddot{x}_{5ref} + g)}{\cos x_7 \cos x_9} \quad (3.24)$$

Sehingga didapatkan persamaan sinyal kontrol ekuivalen U_1 adalah

$$U_{1eq} = \frac{m(\tau \dot{e}_5 + \ddot{x}_{5ref} + g)}{\cos x_7 \cos x_9} \quad (3.25)$$

Pada pengujian, akan dibuat dua jenis sinyal kontrol natural, yaitu sinyal kontrol diskontinu dan sinyal kontrol kontinyu berupa fungsi saturasi dan struktur PID.

Pada tahapan pertama akan dilakukan desain sinyal kontrol natural diskontinu sebagai berikut.

Didefinisikan fungsi Lyapunov pada Persamaan (3.26)

$$V_i = \frac{1}{2} s^2 \quad (3.26)$$

Dimana S adalah permukaan luncur (*sliding surface*) yang diinginkan. Turunan pertama fungsi Lyapunov adalah

$$\dot{s} < 0 \quad (3.27)$$

Pada *Sliding Mode Control*, kondisi pada Persamaan (3.27) harus terpenuhi untuk menjamin sistem stabil.

Berdasarkan syarat kestabilan Lyapunov, maka dipilih :

$$\dot{s} = - \frac{(\cos x_7 \cos x_9) U_N}{m} \quad (3.28)$$

Dipilih sinyal kontrol natural sebagai berikut :

$$U_N = \frac{m.a.sign(s)}{\cos x_7 \cos x_9} \quad (3.29)$$

Dimana konstanta $a > 0$, sehingga untuk nilai $S > 0$, maka $\dot{s} < 0$, dan sebaliknya.

Persamaan sinyal kontrol natural pada (3.29) akan menghasilkan fenomena *switching* berfrekuensi tinggi yang disebut *chattering*. Fenomena ini akan menyebabkan kerusakan pada elemen kontroler,

terutama elemen mekanik. Oleh karena itu, untuk mengurangi efek dari *chattering*, maka fungsi signum pada Persamaan (3.29) diganti dengan fungsi saturasi sebagai berikut :

$$U_N = \frac{m.a.sat(s)}{\cos x_7 \cos x_9} \quad (3.30)$$

Pada tahapan kedua, akan dirancang sinyal kontrol natural dengan mengikuti struktur kontroler PID. Sinyal kontrol ini, selain dapat mengurangi *chattering*, juga dapat mencapai *sliding surface* dengan lebih cepat, dan dapat mengurangi *overshoot*.

Berdasarkan Persamaan (3.28), dipilih sinyal kontrol natural sebagai berikut :

$$U_N = \frac{m(k_p s + k_i \int s dt + k_d \frac{ds}{dt})}{\cos x_7 \cos x_9} \quad (3.31)$$

Dimana nilai $k_p, k_i, k_d > 0$, sehingga untuk mencapai kestabilan Lyapunov seperti pada Persamaan (3.27), maka kondisi berikut harus terpenuhi :

$$\dot{s}\dot{s} < 0$$

$$-s(k_p s + k_i \int s dt + k_d \frac{ds}{dt}) < 0 \quad (3.32)$$

Dimana nilai k_p, k_i, k_d di-*tuning* secara manual untuk memenuhi kondisi kestabilan Lyapunov, dengan syarat $k_p, k_i, k_d > 0$.

Dari persamaan sinyal kontrol ekuivalen dan sinyal kontrol natural yang mendeskripsikan ketinggian dari *quadcopter*, diperoleh persamaan U_1 seperti pada Persamaan (3.33).

$$U_1 = U_{leq} + U_{IN} \quad (3.33)$$

Untuk sinyal kontrol natural menggunakan fungsi signum, didapatkan sinyal kontrol seperti pada Persamaan (3.34).

$$U_1 = \frac{m(\tau \dot{e}_s + \ddot{x}_{sref} + g) + a(sign(s_s))}{\cos x_7 \cos x_9} \quad (3.34)$$

Untuk sinyal kontrol natural menggunakan fungsi saturasi, didapatkan sinyal kontrol seperti pada Persamaan (3.35).

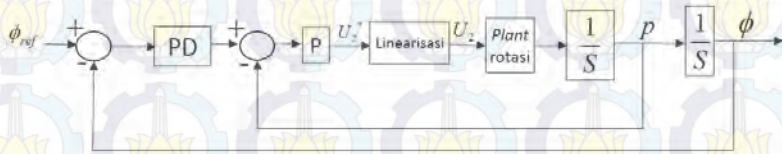
$$U_1 = \frac{m(\tau \dot{e}_s + \ddot{x}_{sref} + g) + a(sat(s_s))}{\cos x_7 \cos x_9} \quad (3.35)$$

Untuk sinyal kontrol natural menggunakan struktur PID, didapatkan sinyal kontrol seperti pada Persamaan (3.36).

$$U_1 = \frac{m(\tau \dot{e}_s + \ddot{x}_{s_{ref}} + g) + k_{pz}s_5 + k_{iz} \int s_5 dt + k_{dz} \frac{ds_5}{dt}}{\cos x_7 \cos x_9} \quad (3.36)$$

3.7.2. Pengaturan Sudut Roll

Sudut *roll* pada *quadcopter* merupakan sudut yang menyebabkan *quadcopter* melintas di sepanjang sumbu y. Pengaturan yang akan dirancang merupakan pengaturan dengan masukan berupa sudut *roll* referensi yang dihasilkan oleh sinyal kontrol gerak translasi sumbu x dan sumbu y, yaitu U_x dan U_y . Sudut *roll* diharapkan akan mampu melakukan proses *tracking* terhadap referensi sudut *roll* yang berubah-ubah sesuai dengan gerakan *quadcopter*. Diagram blok pengaturan sudut *roll* ditunjukkan pada Gambar 3.5.



Gambar 3.5 Diagram Blok Pengaturan Sudut Roll

Dari diagram blok pengaturan sudut *roll*, dapat dilihat bahwa sinyal kontrol yang mempengaruhi sudut *roll* adalah U_2 . Pengaturan sudut *roll* dilakukan dengan mengatur dinamika rotasi pada *quadcopter*. Sudut yang diinginkan adalah sesuai dengan referensi yang berubah-ubah mengikuti trajektori dari *quadcopter*.

Pengaturan posisi sudut *roll* dilakukan dengan menggunakan kontroler PD untuk pengaturan posisi, dan kontroler P untuk pengaturan kecepatan. Pada kedua sistem ini kemudian akan dilakukan proses *cascade*, dengan pengaturan kecepatan sebagai *loop* dalam (*inner loop*), dan pengaturan posisi sebagai *loop* luar (*outer loop*). Untuk mendesain kontroler PD dan P, terlebih dahulu dilakukan proses *feedback linearization* seperti yang telah dijelaskan pada Bab II.

Berikut akan dijabarkan perancangan *PD-P Cascade* untuk sudut *roll*.

Persamaan (3.5) bagian rotasi dapat dituliskan sebagai berikut :

$$\dot{p} = -0,5495qr - 0,00017q\Omega + 0,2052U_2 \quad (3.37)$$

$$\dot{q} = 0,1675pr - 0,0094p\Omega + 2,955U_3 \quad (3.38)$$

$$\dot{r} = -2,0257pq + 0,0954U_4 \quad (3.39)$$

Akan dilakukan pengaturan posisi sudut *roll*, sehingga Persamaan (3.37) menjadi sebagai berikut :

$$\dot{p} = -0,5495qr - 0,00017q\Omega + 0,2052U_2 \quad (3.37)$$

$$\dot{p} = U_2^*, \text{ dimana}$$

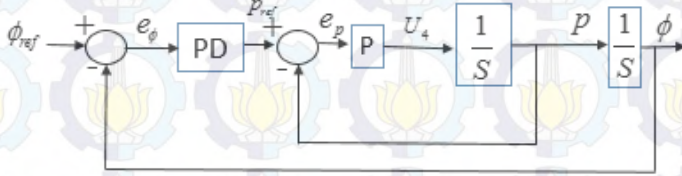
$$U_2^* = -0,5495qr - 0,00017q\Omega + 0,2052U_2 \quad (3.40)$$

Maka, didapatkan

$$p = \frac{1}{S} \dot{p} \quad (3.41)$$

$$\phi = \frac{1}{S} p \quad (3.42)$$

Dimana p adalah kecepatan sudut *roll*, dan ϕ adalah posisi sudut *roll*. Diagram blok untuk desain kontroler ditunjukkan pada Gambar 3.6.



Gambar 3.6 Diagram Blok Desain Kontroler Pengaturan Sudut *Roll*

Persamaan pada *inner loop* ditunjukkan pada Persamaan (3.43).

$$\frac{p}{p_{ref}} = \frac{1}{\tau^* S + 1} \quad (3.43)$$

Diinginkan *transfer function* sistem $\frac{1}{\tau^* S + 1}$, sehingga Persamaan

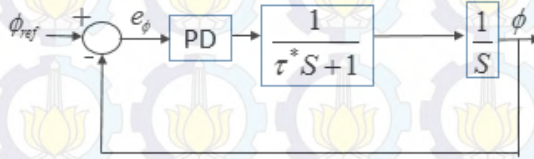
(3.43) menjadi sebagai berikut :

$$\frac{\frac{k_p}{S}}{\frac{k_p}{S} + 1} = \frac{1}{\tau^* S + 1} \quad (3.44)$$

$$\frac{1}{\frac{1}{k_p}S+1} = \frac{1}{\tau^*S+1} \quad (3.45)$$

$$\tau^* = \frac{1}{k_p} \quad (3.46)$$

Sedangkan, diagram blok untuk desain kontroler pada *outer loop* adalah sebagai berikut :



Gambar 3.7. Diagram Blok Desain Kontroler Pengaturan Sudut Roll (*Loop Luar*)

Misalkan struktur pada kontroler PD adalah $u = k_p(\tau_d S + 1)$, didapatkan Persamaan (3.47).

$$\frac{\phi}{\phi_{ref}} = \frac{\frac{k_p}{S}(\tau_d S + 1)(\frac{1}{\tau^* S + 1})}{1 + \frac{k_p}{S}(\tau_d S + 1)(\frac{1}{\tau^* S + 1})} \quad (3.47)$$

Diinginkan *transfer function* sistem $\frac{1}{\tau^{**}S+1}$, sehingga Persamaan (3.47) dapat dinyatakan seperti pada Persamaan (3.48)

$$\frac{1}{\tau^{**}S+1} = \frac{\frac{k_p}{S}(\tau_d S + 1)(\frac{1}{\tau^* S + 1})}{1 + \frac{k_p}{S}(\tau_d S + 1)(\frac{1}{\tau^* S + 1})} \quad (3.48)$$

Dipilih $\tau_d = \tau^*$, sehingga didapatkan Persamaan (3.49)-(3.51).

$$\frac{1}{\tau^{**}S+1} = \frac{\frac{k_p}{S}}{1 + \frac{k_p}{S}} \quad (3.49)$$

$$\frac{1}{\tau^{**}S+1} = \frac{1}{\frac{1}{k_p}S+1} \quad (3.50)$$

$$k_p = \frac{1}{\tau^{**}} \quad (3.51)$$

Pada *loop* dalam (*inner loop*), diinginkan *time constant* (τ^*) respon kecepatan sudut *roll* (p) sebesar 0,1 s, maka didapatkan :

$$k_{p1} = \frac{1}{\tau^*} = 10 \quad (3.43)$$

Selanjutnya akan dilakukan perancangan kontroler PD pada *loop* luar (*outer loop*), yaitu pengaturan posisi sudut *roll*. Untuk *outer loop*, *time constant* yang diinginkan minimal harus 3 kali lebih besar dari *time constant* pada *inner loop*.

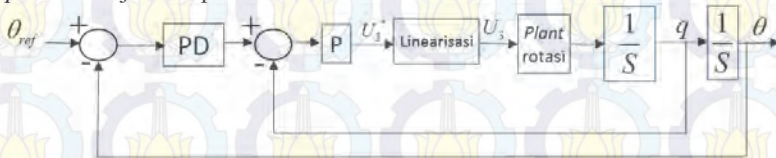
Diinginkan *time constant* (τ^{**}) respon posisi sudut *roll* sebesar 0,3 s, dan dipilih $\tau_d = \tau^{**}$, maka didapatkan :

$$k_{p2} = \frac{1}{\tau^{**}} = \frac{1}{0,3} \quad (3.45)$$

$$\tau_d = \tau^* = 0,1 \quad (3.46)$$

3.7.3. Pengaturan Sudut *Pitch*

Sudut *pitch* pada *quadcopter* merupakan sudut yang menyebabkan *quadcopter* melintas di sepanjang sumbu x. Pengaturan yang akan dirancang merupakan pengaturan dengan masukan berupa sudut *pitch* referensi yang dihasilkan oleh sinyal kontrol gerak translasi sumbu x dan sumbu y, yaitu U_x dan U_y . Sudut *pitch* diharapkan akan mampu melakukan proses *tracking* terhadap referensi sudut *pitch* yang berubah-ubah sesuai dengan gerakan *quadcopter*. Diagram blok pengaturan sudut *pitch* ditunjukkan pada Gambar 3.6.



Gambar 3.8 Diagram Blok Pengaturan Sudut *Pitch*

Dengan cara perancangan yang sama seperti kontrol sudut *roll*, akan diperoleh konstanta k_{p1} , k_{p2} , dan τ_d untuk pengendalian sudut *pitch*. Untuk memperoleh konstanta tersebut, maka model matematika dari sistem harus dilinerisasi pada titik tertentu. Dipilih metode linearisasi berupa *feedback linearization*. Persamaan untuk mendapatkan nilai k_{p1} , k_{p2} , dan τ_d ditunjukkan oleh persamaan-persamaan yang disajikan pada subbab ini.

Dari diagram blok pengaturan sudut *pitch*, dapat dilihat bahwa sinyal kontrol yang mempengaruhi sudut *pitch* adalah U_3 . Pengaturan sudut *pitch* dilakukan dengan mengatur dinamika rotasi pada *quadcopter*. Sudut *pitch* yang diinginkan untuk gerak *waypoint tracking* adalah mengikuti sudut *pitch* referensi yang dipengaruhi oleh U_x dan U_y . Berikut akan dijabarkan perancangan *PD-P Cascade* untuk sudut *pitch*.

Persamaan (3.5) bagian rotasi dapat dituliskan sebagai berikut :

$$\dot{p} = -0,5495qr - 0,00017q\Omega + 0,2052U_2 \quad (3.37)$$

$$\dot{q} = 0,1675pr - 0,0094p\Omega + 2,955U_3 \quad (3.38)$$

$$\dot{r} = -2,0257pq + 0,0954U_4 \quad (3.39)$$

Akan dilakukan pengaturan posisi sudut *pitch*, sehingga Persamaan (3.38) menjadi sebagai berikut :

$$\dot{q} = 0,1675pr - 0,0094p\Omega + 2,955U_3 \quad (3.37)$$

$$\dot{q} = U_3^*, \text{ dimana}$$

$$U_3^* = 0,1675pr - 0,0094p\Omega + 2,955U_3 \quad (3.47)$$

Maka, didapatkan

$$q = \frac{1}{S} \dot{q} \quad (3.48)$$

$$\theta = \frac{1}{S} q \quad (3.49)$$

Pada *loop* dalam (*inner loop*), diinginkan *time constant* (τ^*) respon kecepatan sudut *pitch* (q) sebesar 0,1 s, maka didapatkan :

$$k_{p1} = \frac{1}{\tau^*} = 10 \quad (3.50)$$

Selanjutnya akan dilakukan perancangan kontroler PD pada *loop* luar (*outer loop*), yaitu pengaturan posisi sudut *pitch*. Untuk *outer loop*,

time constant yang diinginkan minimal harus 3 kali lebih besar dari *time constant* pada *inner loop*.

Diinginkan *time constant* (τ^{**}) respon posisi sudut pitch sebesar 1 s, dan dipilih $\tau_d = \tau^{**}$, maka didapatkan :

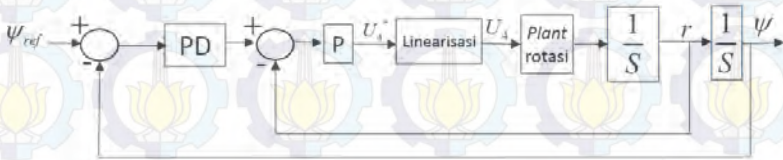
$$\frac{\theta}{\theta_{ref}} = \frac{\frac{k_{p2}}{S}}{1 + \frac{k_{p2}}{S}} = \frac{1}{\frac{1}{k_{p2}}S + 1} = \frac{1}{\tau^*S + 1} \quad (3.51)$$

$$k_{p2} = \frac{1}{\tau^{**}} = \frac{1}{0,3} \quad (3.52)$$

$$k_{d2} = \tau_d = \tau^* = 0,1 \quad (3.53)$$

3.7.4. Pengaturan Sudut Yaw

Tidak seperti sudut *pitch* dan sudut *roll*, Sudut *yaw* pada *quadcopter* tidak mempengaruhi gerakan *quadcopter* baik di sumbu x maupun sumbu y. Oleh karena itu, pengaturan yang akan dirancang merupakan pengaturan dengan masukan nol, karena diharapkan *quadcopter* dapat melakukan gerak *vertical take-off* dan *landing* dengan stabil dan tidak mengganggu proses *tracking*. Diagram blok pengaturan sudut *yaw* ditunjukkan pada Gambar 3.9.



Gambar 3.9 Diagram Blok Pengaturan Sudut Yaw

Dengan cara perancangan yang sama seperti kontrol sudut *roll* dan *pitch*, akan diperoleh konstanta k_{p1} , k_{p2} , dan τ_d untuk pengendalian sudut *yaw*. Untuk memperoleh konstanta tersebut, maka model matematika dari sistem harus dilinerisasi pada titik tertentu. Akan digunakan metode linearisasi berupa *feedback linearization*. Persamaan untuk mendapatkan nilai k_{p1} , k_{p2} , dan τ_d ditunjukkan oleh persamaan-persamaan yang disajikan pada subbab ini.

Dari diagram blok pengaturan sudut *yaw*, dapat dilihat bahwa sinyal kontrol yang mempengaruhi sudut *yaw* adalah U_4 . Pengaturan sudut *yaw* dilakukan dengan mengatur dinamika rotasi pada *quadcopter*. Sudut *yaw* yang diinginkan untuk gerak *waypoint tracking* adalah nol. Berikut akan dijabarkan perancangan *PD-P Cascade* untuk sudut *yaw*.

Persamaan (3.5) bagian rotasi dapat dituliskan sebagai berikut :

$$\dot{p} = -0,5495qr - 0,00017q\Omega + 0,2052U_2 \quad (3.37)$$

$$\dot{q} = 0,1675pr - 0,0094p\Omega + 2,955U_3 \quad (3.38)$$

$$\dot{r} = -2,0257pq + 0,0954U_4 \quad (3.39)$$

Akan dilakukan pengaturan posisi sudut *yaw*, sehingga Persamaan (3.39) menjadi sebagai berikut :

$$\dot{r} = -2,0257pq + 0,0954U_4 \quad (3.39)$$

$\dot{r} = U_4^*$, dimana

$$U_4^* = \frac{1}{0,0954} (-2,0257pq + 0,0954U_4) \quad (3.54)$$

Maka, didapatkan

$$r = \frac{1}{S} \dot{r} \quad (3.55)$$

$$\psi = \frac{1}{S} r \quad (3.56)$$

Pada *loop* dalam (*inner loop*), diinginkan *time constant* (τ^*) respon kecepatan sudut *yaw* (r) sebesar 0,1 s, maka didapatkan :

$$k_{p1} = \frac{1}{\tau^*} = 10 \quad (3.57)$$

Selanjutnya akan dilakukan perancangan kontroler PD pada *loop* luar (*outer loop*), yaitu pengaturan posisi sudut *yaw*. Untuk *outer loop*, *time constant* yang diinginkan minimal harus 3x lebih besar dari *time constant* pada *inner loop*.

Diinginkan *time constant* (τ^{**}) respon posisi sudut *yaw* sebesar 0,3 s, dan dipilih $\tau_d = \tau^{**}$, maka didapatkan :

$$\frac{\psi}{\psi_{ref}} = \frac{\frac{k_{p2}}{S}}{1 + \frac{k_{p2}}{S}} = \frac{1}{\frac{1}{k_{p2}}S + 1} = \frac{1}{\tau^*S + 1} \quad (3.58)$$

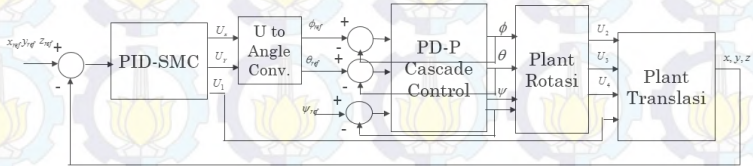
$$k_{p2} = \frac{1}{\tau^{**}} = \frac{1}{0,3} \quad (3.59)$$

$$\tau_d = \tau^* = 0,1 \quad (3.60)$$

3.7.5. Pengaturan Posisi Sumbu X

Desain kontroler pengaturan posisi *quadcopter* pada sumbu x menggunakan cara yang sama dengan desain kontroler posisi *quadcopter* di sumbu z.

Diagram blok sistem pengaturan posisi *quadcopter* pada sumbu x dan sumbu y ditunjukkan pada Gambar 3.10.



Gambar 3.10 Diagram Blok Pengaturan Posisi *Quadcopter*

Untuk mendapatkan sinyal kontrol ekuivalen (U_{eq}), maka didefinisikan *error tracking* yang merupakan perbedaan antara nilai posisi *quadcopter* dengan posisi referensi sebagai berikut :

$$e_1 = x_{1ref} - x_1 \quad (3.61)$$

Dengan x_{1ref} adalah sinyal referensi. Selanjutnya akan dirancang permukaan luncur (*sliding surface*), yang didefinisikan sebagai berikut :

$$s_1 = \tau e_1 + \dot{e}_1 \quad (3.62)$$

Untuk membuat permukaan luncur menuju 0 pada waktu tak hingga, maka s diturunkan terhadap waktu, sehingga didapatkan

$$\dot{s}_1 = \tau \dot{e}_1 + \ddot{e}_1 = 0 \quad (3.63)$$

Substitusi e_1 dengan Persamaan (3.61), maka didapatkan Persamaan (3.64).

$$\dot{s}_1 = \tau \dot{e}_1 - \ddot{x}_1 + \ddot{x}_{1ref} = 0 \quad (3.64)$$

Dengan $\ddot{x}_1 = \ddot{x}_2$ maka didapatkan Persamaan (3.65).

$$\dot{s}_1 = \tau \dot{e}_1 - \ddot{x}_2 + \ddot{x}_{1ref} = 0 \quad (3.65)$$

Substitusi dengan Persamaan (3.9), didapatkan Persamaan (3.66).

$$\dot{s}_1 = \tau \dot{e}_1 - \frac{U_1}{m} U_x + \ddot{x}_{1ref} = 0 \quad (3.66)$$

Sehingga didapatkan persamaan sinyal kontrol ekuivalen U_{xeq} seperti pada Persamaan (3.67).

$$U_{xeq} = \frac{m}{U_1} (\ddot{x}_{1ref} + \tau \dot{e}_1) \quad (3.67)$$

Selanjutnya akan dirancang sinyal kontrol natural U_N . Prosedur perancangan adalah sama dengan perancangan sinyal kontrol natural untuk mengatur posisi *quadcopter* di sumbu z.

Dari persamaan sinyal kontrol ekuivalen dan sinyal kontrol natural yang mendeskripsikan posisi *quadcopter* pada sumbu x, diperoleh persamaan U_x seperti pada Persamaan (3.68).

$$U_x = U_{xeq} + U_{xN} \quad (3.68)$$

Untuk sinyal kontrol natural menggunakan fungsi signum, didapatkan sinyal kontrol seperti pada Persamaan (3.69).

$$U_x = \frac{m}{U_1} ((\ddot{x}_{1ref} + \tau \dot{e}_1) + a(\text{sign}(s_1))) \quad (3.69)$$

Untuk sinyal kontrol natural menggunakan fungsi saturasi, didapatkan sinyal kontrol seperti pada Persamaan (3.70).

$$U_x = \frac{m}{U_1} ((\ddot{x}_{1ref} + \tau \dot{e}_1) + a(\text{sat}(s_1))) \quad (3.70)$$

Untuk sinyal kontrol natural menggunakan struktur PID, didapatkan sinyal kontrol seperti pada Persamaan (3.71).

$$U_x = \frac{m}{U_1} ((\ddot{x}_{1ref} + \tau \dot{e}_1) + k_{px} s_1 + k_{ix} \int s_1 dt + k_{dx} \frac{ds_1}{dt}) \quad (3.71)$$

3.7.6. Pengaturan Posisi Sumbu Y

Desain kontroler pengaturan posisi *quadcopter* pada sumbu y menggunakan cara yang sama dengan desain kontroler posisi *quadcopter* di sumbu x dan z. Untuk mendapatkan sinyal kontrol ekuivalen (U_{eq}), maka didefinisikan *error tracking* yang merupakan perbedaan antara nilai posisi *quadcopter* dengan posisi referensi seperti pada Persamaan (3.72).

$$e_3 = x_{3ref} - x_3 \quad (3.72)$$

Dengan x_{3ref} adalah sinyal referensi. Selanjutnya akan dirancang permukaan luncur (*sliding surface*), yang didefinisikan seperti pada Persamaan (3.73).

$$s_3 = \tau e_3 + \dot{e}_3 \quad (3.73)$$

Untuk membuat permukaan luncur menuju 0 pada waktu tak hingga, maka s diturunkan terhadap waktu, sehingga didapatkan Persamaan (3.74).

$$\dot{s}_3 = \tau \dot{e}_3 + \ddot{e}_3 = 0 \quad (3.74)$$

Substitusi e_3 dengan Persamaan (3.74), maka didapatkan Persamaan (3.75).

$$\dot{s}_3 = \tau \dot{e}_3 - \ddot{x}_3 + \ddot{x}_{3ref} = 0 \quad (3.75)$$

Dengan $\ddot{x}_3 = \dot{x}_4$, maka didapatkan Persamaan (3.76).

$$\dot{s}_3 = \tau \dot{e}_3 + \dot{x}_4 - \ddot{x}_{3ref} = 0 \quad (3.76)$$

Substitusi dengan Persamaan (3.9), didapatkan Persamaan (3.77).

$$\dot{s}_3 = -\tau \dot{e}_3 + \frac{U_1}{m} U_Y - \ddot{x}_{3ref} = 0 \quad (3.77)$$

Sehingga didapatkan persamaan sinyal kontrol ekuivalen U_{Yeq} seperti pada Persamaan (3.78).

$$U_Y = \frac{m}{U_1} (\ddot{x}_{3ref} + \tau \dot{e}_3) \quad (3.78)$$

Selanjutnya akan dirancang sinyal kontrol natural U_N . Prosedur perancangan adalah sama dengan perancangan sinyal kontrol natural untuk mengatur posisi *quadcopter* di sumbu z dan x.

Dari persamaan sinyal kontrol ekuivalen dan sinyal kontrol natural yang mendeskripsikan posisi *quadcopter* pada sumbu y, diperoleh persamaan U_y seperti pada Persamaan (3.79).

$$U_Y = U_{Yeq} + U_{YN} \quad (3.79)$$

Untuk sinyal kontrol natural menggunakan fungsi signum, didapatkan sinyal kontrol seperti pada Persamaan (3.80).

$$U_Y = \frac{m}{U_1} ((\ddot{x}_{3ref} + \tau \dot{e}_3) + a(\text{sign}(s_3))) \quad (3.80)$$

Untuk sinyal kontrol natural menggunakan fungsi saturasi, didapatkan sinyal kontrol seperti pada Persamaan (3.81).

$$U_Y = \frac{m}{U_1} ((\ddot{x}_{3ref} + \tau \dot{e}_3) + a(\text{sat}(s_3))) \quad (3.81)$$

Untuk sinyal kontrol natural menggunakan struktur PID, didapatkan sinyal kontrol seperti pada Persamaan (3.82).

$$U_Y = \frac{m}{U_1} ((\ddot{x}_{3ref} + \tau \dot{e}_3) + k_{py}s_3 + k_{iy} \int s_3 dt + k_{dy} \frac{ds_3}{dt}) \quad (3.82)$$

3.8. Konversi Sinyal Kontrol Sumbu X dan Y ke Dalam Sudut Referensi

Sinyal kontrol U_X dan U_Y yang telah didapatkan, akan diubah ke dalam sudut *roll* dan *pitch* referensi. Sudut roll dan pitch referensi tersebut selanjutnya akan masuk ke sistem kontrol rotasi yang telah dijelaskan diatas. Berikut akan dijabarkan perancangan konversi U_X dan U_Y ke dalam ϕ_{ref} dan θ_{ref} .

Untuk mengubah nilai U_X dan U_Y yang merupakan nilai sudut rotasi referensi ke dalam nilai referensi sudut ϕ dan θ , dengan memisalkan sudut *yaw* (ψ) = 0, maka Persamaan (3.10) dan (3.11) dapat dinyatakan seperti Persamaan (3.83) dan (3.84).

$$U_X = \cos \phi \sin \theta \quad (3.83)$$

$$U_Y = -\sin \phi \quad (3.84)$$

Sehingga untuk mendapatkan ϕ_{ref} dan θ_{ref} , maka didapatkan Persamaan (3.85) dan (3.86).

$$\phi_{ref} = -\sin^{-1}(U_Y) \quad (3.85)$$

$$\theta_{ref} = \sin^{-1}\left(\frac{U_X}{\cos \phi}\right) \quad (3.86)$$

BAB IV

HASIL DAN ANALISIS

Pada bab ini, dipaparkan hasil simulasi dan analisis dari *Sliding Mode Control* yang telah dirancang pada Bab III pada *plant quadcopter*.

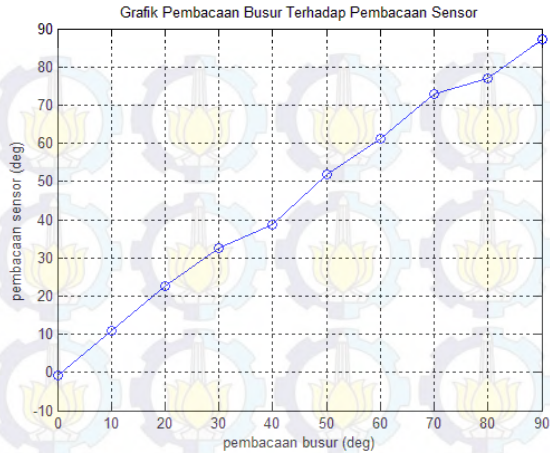
4.1. Pengujian Sensor

Pada tahapan ini dilakukan pengujian sensor. Sensor yang digunakan yaitu *gyro* dan *accelerometer*. Dilakukan pengujian terhadap sudut *roll*, *pitch*, dan *yaw*, serta ketinggian dari *quadcopter* untuk menguji akurasi dari sensor. Hasil pengujian ditunjukkan pada Tabel 4.1-Tabel 4.3.

Tabel 4.1 Pengujian Sudut Roll

No	Pembacaan Busur (°)	Pembacaan Sensor (°)
1	0	-0,68
2	10	10,82
3	20	22,7
4	30	32,6
5	40	38,7
6	50	51,8
7	60	61,3
8	70	72,8
9	80	77,06
10	90	87,1

Dari Tabel 4.1, dapat dilihat bahwa sensor mampu mengukur sudut *roll* dengan *error RMS* sebesar 4,8022%. Grafik pembacaan sensor terhadap pembacaan busur adalah seperti pada Gambar 4.1.



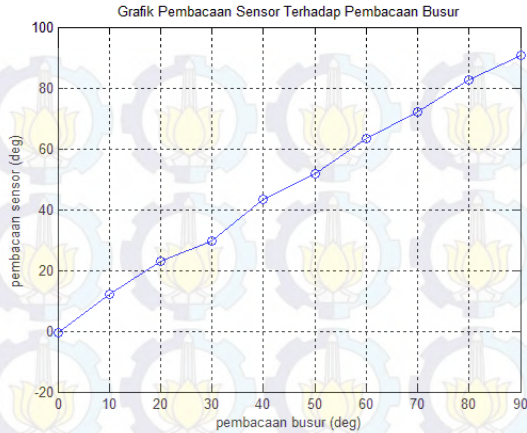
Gambar 4.1. Pengujian Sudut *Roll*

Sedangkan, untuk pengujian sudut *pitch* ditampilkan pada Tabel 4.2.

Tabel 4.2. Pengujian Sudut *Pitch*

No	Pembacaan Busur (°)	Pembacaan Sensor (°)
1	0	-0,38
2	10	12
3	20	23,1
4	30	29,67
5	40	43,2
6	50	51,8
7	60	63,2
8	70	72
9	80	82,46
10	90	90,7

Dari Tabel 4.2, dapat dilihat bahwa sensor mampu mengukur sudut *pitch* dengan *error* RMS 4,8750%. Grafik pembacaan sensor terhadap pembacaan busur adalah seperti pada Gambar 4.2.



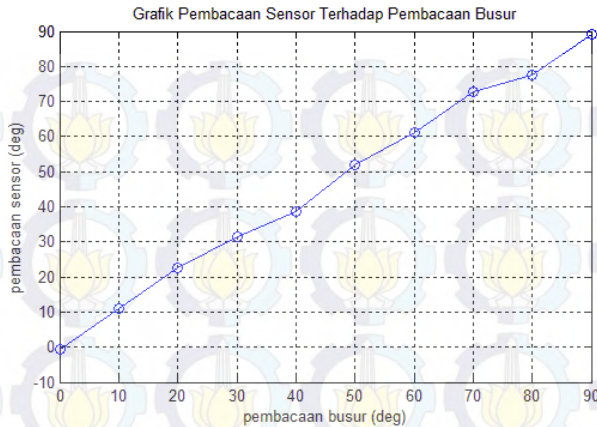
Gambar 4.2. Pengujian Sudut *Pitch*

Sedangkan, untuk pengujian sudut *yaw* ditampilkan pada Tabel 4.3.

Tabel 4.3. Pengujian Sudut *Yaw*

No	Pembacaan Busur (°)	Pembacaan Sensor (°)
1	0	-0,68
2	10	10.82
3	20	22,7
4	30	31,4
5	40	38,8
6	50	52,0
7	60	61,3
8	70	72,8
9	80	77,53
10	90	89,1

Dari Tabel 4.3, dapat dilihat bahwa sensor mampu mengukur sudut *yaw* dengan *error RMS* sebesar 3,9917%. Grafik pembacaan sensor terhadap pembacaan busur adalah seperti pada Gambar 4.3.



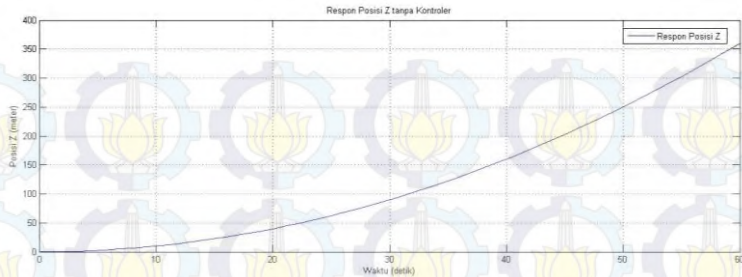
Gambar 4.3. Pengujian Sudut Yaw

4.2. Simulasi

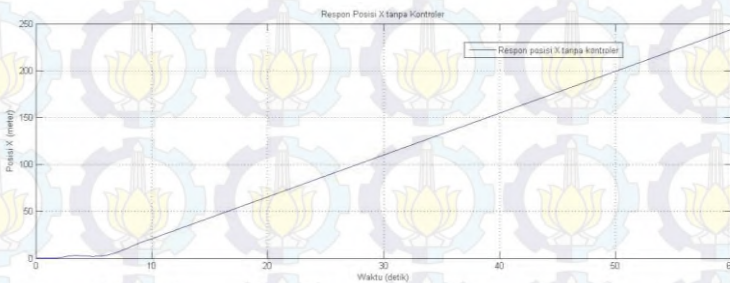
Simulasi sistem merupakan tahap awal yang dilakukan sebelum melakukan implementasi agar dapat diketahui performansi sistem yang akan terjadi pada *plant quadcopter*. Dengan menggunakan simulasi, kesalahan dalam mendesain metode kontrol tidak akan merusak *plant* karena tidak menggunakan *plant* secara langsung. Simulasi pengaturan gerak *waypoint tracking* pada *quadcopter* dilakukan dengan menggunakan perangkat lunak MATLAB. Hal yang akan diuji pada simulasi ini adalah pengujian metode yang digunakan terhadap parameter sistem yang konstan, gangguan pada sudut, dan gangguan pada ketinggian, serta *tracking* mengikuti pola tertentu.

4.2.1. Respon Open Loop

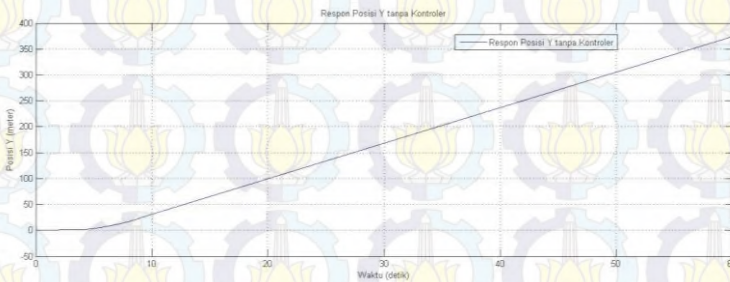
Respon *open loop* dilakukan untuk menguji respon *quadcopter* apabila tidak dilakukan proses kontrol secara *closed-loop*. Respon posisi tanpa kontroler pada *quadcopter* adalah sistem yang tidak stabil. Hal ini ditunjukkan pada Gambar 4.4 dan 4.5, dan 4.6.



Gambar 4.4 Respon Posisi *Quadcopter* Sumbu Z Tanpa Kontroler



Gambar 4.5 Respon Posisi *Quadcopter* Sumbu X Tanpa Kontroler



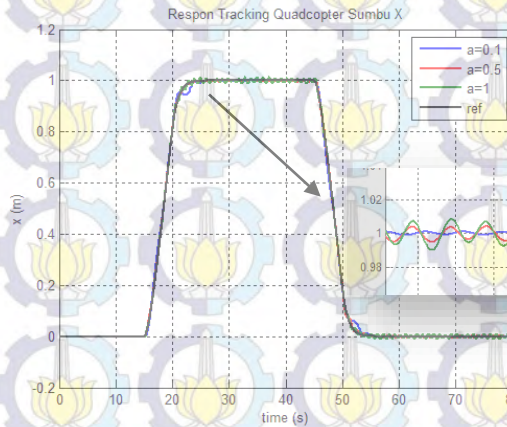
Gambar 4.6 Respon Posisi *Quadcopter* Sumbu Y Tanpa Kontroler

4.2.2. Simulasi *Sliding Mode Control* dengan Fungsi Signum untuk Gerak Waypoint Tracking

Pada bab ini, akan dilakukan simulasi untuk menguji performa kontroler *Sliding Mode*. Algoritma kontrol yang digunakan adalah kontroler *Sliding Mode* dengan sinyal kontrol natural U_N berupa sinyal

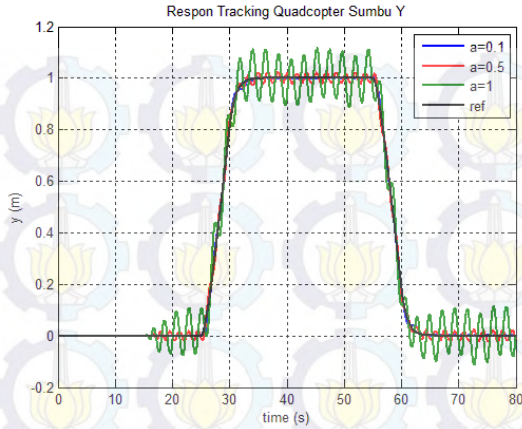
signum. Pada simulasi ini akan dilakukan pengujian gerakan *quadcopter* baik pada sumbu x dan y.

Prosedur pengujiannya adalah dengan menggunakan lintasan *tracking* berbentuk persegi, dengan tujuan untuk menguji respon kontroler posisi x dan y dengan referensi berupa *tracking* untuk nilai τ dan a yang berbeda-beda. Pada kontroler posisi z, digunakan *Sliding Mode Control* dengan fungsi signum, dengan nilai $\tau=1$. Hasil pengujian dengan lintasan berbentuk persegi adalah sebagai berikut :



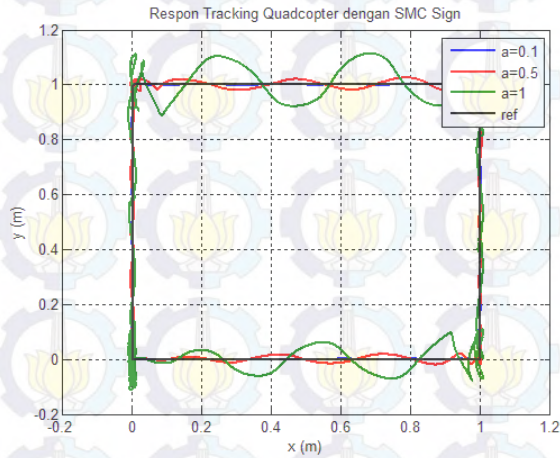
Gambar 4.7 Respon Tracking *Quadcopter* Sumbu X dengan SMC Signum

Respon *quadcopter* pada gerak di sumbu y ditunjukkan pada Gambar 4.8.



Gambar 4.8 Respon *Tracking Quadcopter* Sumbu Y dengan SMC Signum

Sedangkan, lintasan yang dibentuk oleh *quadcopter* ditunjukkan pada Gambar 4.9.



Gambar 4.9 Lintasan *Tracking Quadcopter* dengan SMC Signum

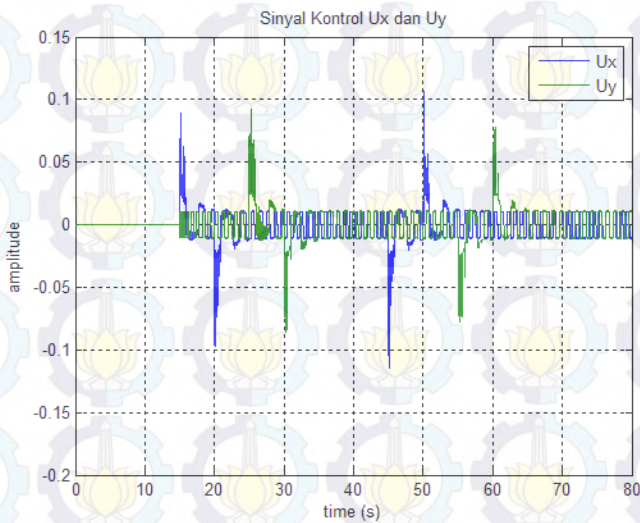
Dari grafik tersebut, didapatkan hasil seperti pada Tabel 4.4.

Tabel 4.4 *Error Tracking Quadcopter* dengan SMC Signum

a_x	a_y	RMSE x	RMSE y
0,1	0,1	2,532%	1,745%
0,5	0,5	0,941%	3,665%
1	1	1,522%	16,087%

Dari tabel tersebut, dapat dilihat bahwa untuk nilai $\tau = 1$ dan nilai a yang bervariasi, nilai a harus dilakukan *tuning* sehingga *quadcopter* dapat melakukan *tracking* dari referensi yang diinginkan dengan *error* seminimal mungkin. Apabila nilai a terlalu besar, maka akan menyebabkan osilasi.

Sinyal kontrol yang dihasilkan oleh kontroler dengan nilai $a = 0,1$ ditunjukkan pada Gambar 4.10.

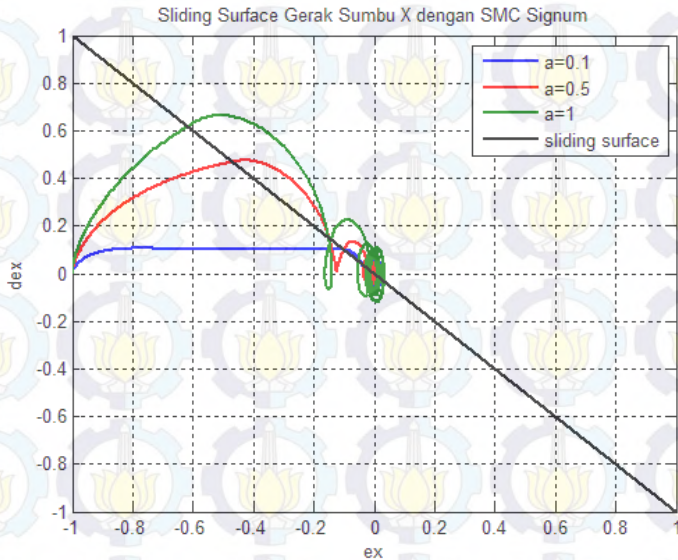


Gambar 4.10 Sinyal Kontrol U_x dan U_y dengan SMC Signum

Dari Gambar 4.10, dapat dilihat bahwa sinyal kontrol mengalami *chattering* atau *switching* dengan frekuensi tinggi. Fenomena *chattering* ini disebabkan oleh unsur diskontinyu dari sistem. Hal ini dapat merusak perangkat yang digunakan, karena mengalami perubahan dalam waktu

yang sangat cepat. Oleh karena itu, pada pengujian selanjutnya fungsi diskontinyu pada fungsi signum diganti dengan fungsi kontinyu.

Untuk mengetahui perilaku dari state sistem pada sliding surface, digunakan *initial condition*. *Initial condition* yang digunakan adalah $x = 1$ m. *Sliding surface* yang dihasilkan oleh SMC dengan fungsi signum ditunjukkan pada Gambar 4.11.



Gambar 4.11 *Sliding Surface* Gerak Sumbu X dengan SMC Signum

Dari Gambar 4.11, dapat dilihat bahwa nilai a yang semakin besar membuat *state error* dari sistem dapat mengikuti *sliding surface* yang telah direncanakan. Namun, semakin besar nilai a , maka semakin besar pula kecenderungan sistem untuk berosilasi. Hal ini dapat dilihat pada Gambar 4.11, dimana SMC dengan nilai $a = 1$ berosilasi di sekitar nilai nol. Oleh karena itu, untuk mendapat respon yang baik, dilakukan *tuning* pada nilai a .

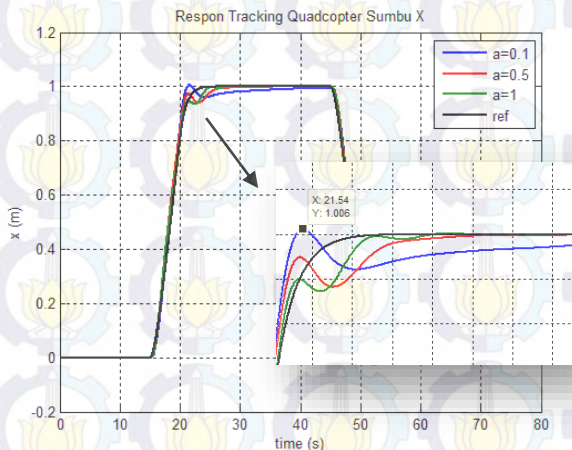
4.2.3. Simulasi *Sliding Mode Control* dengan Fungsi Saturasi Untuk Gerak *Waypoint Tracking*

Pada bab ini, akan dilakukan simulasi untuk menguji performa kontroler *Sliding Mode*. Algoritma kontrol yang digunakan adalah kontroler *Sliding Mode* dengan sinyal kontrol natural U_N berupa sinyal saturasi, dengan tujuan untuk mengurangi *chattering* pada sinyal kontrol. Pada simulasi ini akan dilakukan pengujian gerakan *quadcopter* baik pada sumbu x dan y.

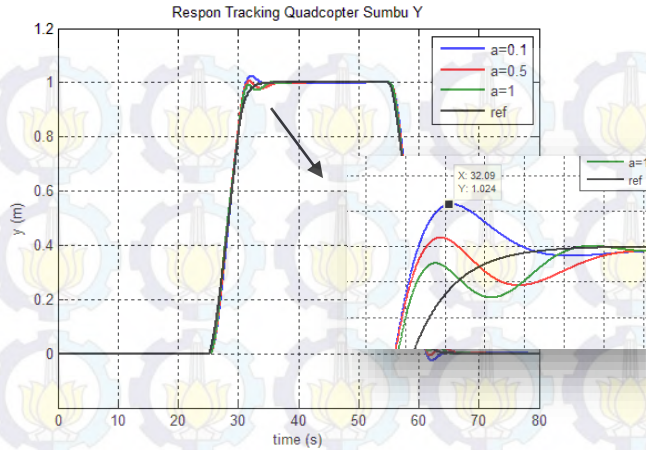
Fungsi saturasi yang digunakan adalah sebagai berikut :

$$\text{sat}(x) = \begin{cases} -1, & x \leq -1 \\ x, & -1 < x < 1 \\ 1, & x > 1 \end{cases} \quad (4.1)$$

Prosedur pengujiannya adalah dengan menggunakan lintasan *tracking* berbentuk persegi, dengan tujuan untuk menguji respon kontroler posisi x dan y dengan referensi berupa *tracking* untuk nilai τ dan a yang berbeda-beda. Pada kontroler posisi z, digunakan *Sliding Mode Control* dengan fungsi saturasi, dengan nilai τ dan a masing-masing 1. Respon sistem ditunjukkan pada Gambar 4.12 dan Gambar 4.13.

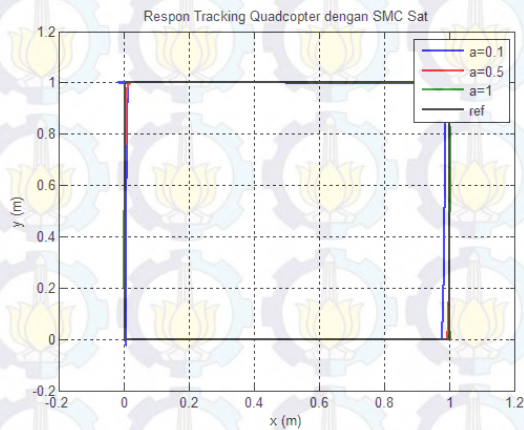


Gambar 4.12 Respon *Tracking Quadcopter* Sumbu X dengan SMC Saturasi



Gambar 4.13 Respon *Tracking Quadcopter SumbuY* dengan SMC Saturasi

Sedangkan, lintasan yang dibentuk oleh *quadcopter* ditunjukkan pada Gambar 4.14.



Gambar 4.14 Lintasan *Tracking Quadcopter* dengan SMC Saturasi

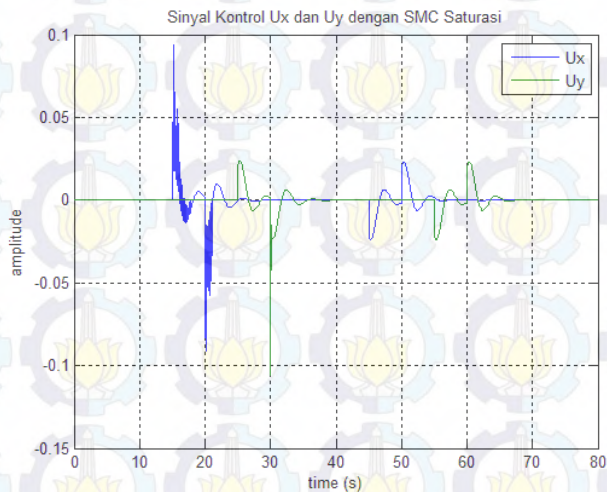
Dari grafik tersebut, didapatkan hasil seperti pada Tabel 4.5.

Tabel 4.5 Error Tracking Quadcopter dengan SMC Saturasi

α_x	α_y	RMSE X	RMSE Y
0,1	0,1	4,364%	4,034%
0,5	0,5	3,204%	3,318%
1	1	2,517%	2,521%

Pada Gambar 4.13, dapat dilihat bahwa *Sliding Mode Control* dengan fungsi saturasi memiliki *overshoot* sebesar 2,4% pada nilai $\alpha = 0,1$. Hal ini disebabkan sinyal kontrol yang dihasilkan lebih kecil. Sinyal kontrol yang lebih besar akan membuat sistem lebih mampu untuk mengikuti trajektori dari *sliding surface* yang direncanakan.

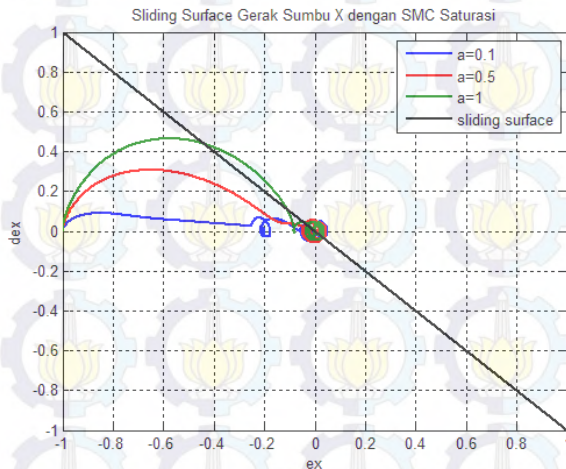
Sinyal kontrol yang dihasilkan oleh SMC saturasi dengan parameter $\alpha = 0,1$ ditampilkan pada Gambar 4.9 dan 4.10.



Gambar 4.15 Sinyal Kontrol U_x dan U_y dengan SMC Saturasi

Dari Gambar 4.15, dapat dilihat bahwa tidak terdapat *chattering* pada sinyal kontrol yang dihasilkan oleh kontroler. Hal ini disebabkan karena unsur diskontinyu pada *signum* dihilangkan dan digantikan oleh saturasi, yang merupakan fungsi kontinyu pada titik-titik tertentu.

Untuk mengetahui perilaku dari state sistem pada sliding surface, digunakan *initial condition*. *Initial condition* yang digunakan adalah $x = 1$ m. *Sliding surface* yang dihasilkan oleh SMC dengan fungsi saturasi adalah sebagai berikut :



Gambar 4.16 *Sliding Surface* Gerak Sumbu X dengan SMC Saturasi

Dari Gambar 4.16, dapat disimpulkan bahwa *Sliding Mode Control* dengan nilai a yang lebih besar akan mampu memaksa *state* dari sistem untuk mengikuti trajektori yang direncanakan.

4.2.4. Simulasi *Sliding Mode Control* dengan Struktur PID untuk Gerak Waypoint Tracking

Pada subbab ini, akan dilakukan simulasi untuk menguji performa kontroler *Sliding Mode*. Algoritma kontrol yang digunakan adalah kontroler *Sliding Mode* dengan sinyal kontrol natural U_N berupa struktur kontroler PID, dengan tujuan untuk mengurangi *chattering* pada sinyal kontrol. Pada simulasi ini akan dilakukan pengujian gerakan *quadcopter* baik pada sumbu x dan y .

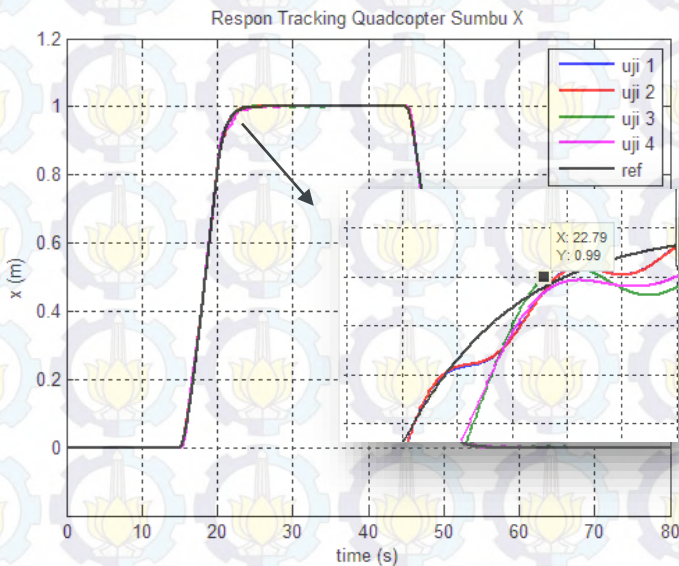
Struktur kontroler PID yang digunakan adalah seperti pada Persamaan (4.2) dan Persamaan (4.3).

$$U_{PID} = k_p(s + \frac{1}{\tau_i} \int s dt + \tau_d \frac{ds}{dt}) \quad (4.2)$$

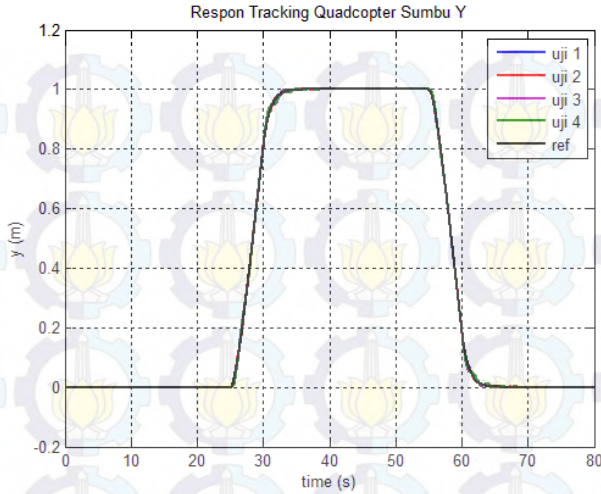
$$U_{PID} = k_p s + k_i \int s dt + k_d \frac{ds}{dt} \quad (4.3)$$

Prosedur pengujiannya adalah dengan menggunakan lintasan *tracking* berbentuk persegi, dengan tujuan untuk menguji respon kontroler posisi x dan y dengan referensi berupa *tracking* untuk nilai τ dan a yang berbeda-beda. Pada kontroler posisi z, digunakan *Sliding Mode Control* dengan fungsi saturasi, dengan nilai τ dan a masing-masing 1.

Hasil pengujian ditampilkan pada Gambar 4.17 dan Gambar 4.18.

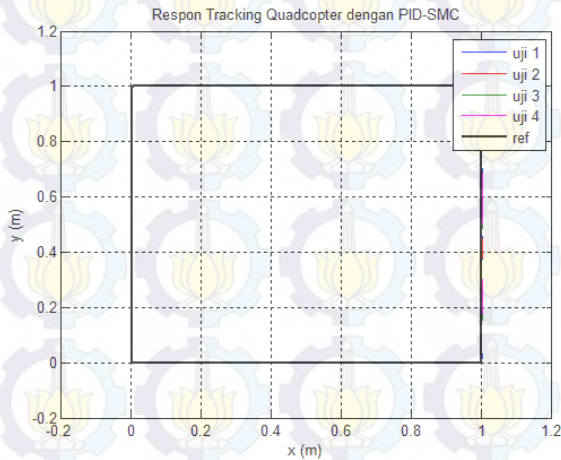


Gambar 4.17 Respon *Tracking Quadcopter Sumbu X* dengan PID-SMC



Gambar 4.18 Respon *Tracking Quadcopter* Sumbu Y dengan PID-SMC

Sedangkan, lintasan yang dibentuk oleh *quadcopter* ditunjukkan pada Gambar 4.19.



Gambar 4.19 Lintasan *Tracking Quadcopter* dengan PID-SMC

Pada Gambar 4.17 dan 4.18, dapat dilihat bahwa *Sliding Mode Control* dengan berbasis PID dengan nilai k_p yang lebih besar akan mempunyai *error* yang lebih kecil, sedangkan k_i yang lebih besar akan membuat sistem memiliki *overshoot* yang lebih besar pula. Pemilihan parameter PID harus dilakukan secara teliti agar selalu memenuhi syarat kestabilan Lyapunov.

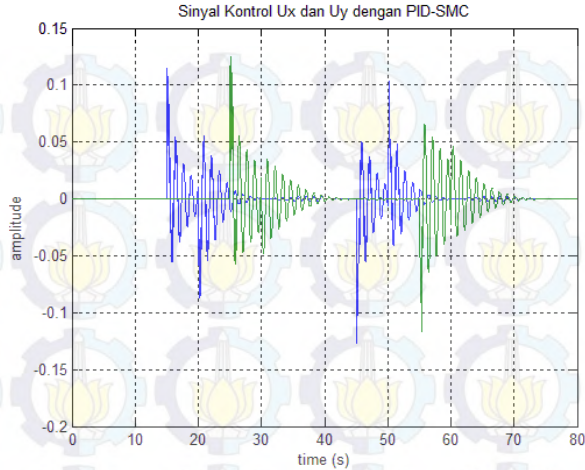
Perbandingan konstanta pada tiap-tiap pengujian ditunjukkan pada Tabel 4.6, dengan parameter pengujian pada gerak sumbu x adalah sama dengan parameter pengujian pada gerak sumbu y.

Tabel 4.6 *Error Tracking Quadcopter* dengan PID-SMC

Uji	k_p	k_i	k_d	RMSE X	RMSE Y
1	15	0,01	0,01	0,2%	0,47%
2	15	0,1	0,01	0,419%	0,527%
3	5	0,1	0,01	1,12%	1,12%
4	5	0,1	0,1	1,02%	0,99%

Dari Tabel 4.7, dapat dilihat bahwa *Sliding Mode Control* dengan sinyal kontrol natural berupa struktur PID mampu melakukan *tracking* terhadap lintasan berbentuk persegi, dengan *error RMS* pada sumbu x sebesar 0,2%, dan *error RMS* pada sumbu y sebesar 0,47% . *Error RMS* tersebut diperoleh dari parameter $k_{px} = 15, k_{ix} = 0,01, k_{dx} = 0,01$, dan $k_{py} = 15, k_{iy} = 0,01, k_{dy} = 0,01$.

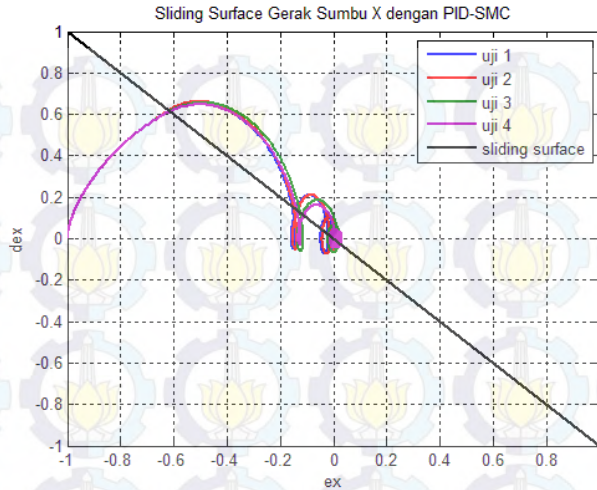
Sinyal kontrol yang dihasilkan oleh Uji 1 PID-SMC dengan parameter seperti pada Tabel 4.6 ditampilkan pada Gambar 4.20.



Gambar 4.20 Sinyal Kontrol U_x dan U_y dengan PID-SMC

Dari Gambar 4.20, dapat dilihat bahwa tidak terdapat *chattering* pada sinyal kontrol, namun terdapat osilasi menuju nilai nol pada saat *quadcopter* bergerak, hal ini dikarenakan kontroler menyesuaikan posisi *set point* dari *quadcopter* yang berubah-ubah.

Untuk mengetahui perilaku dari state sistem pada *sliding surface*, digunakan *initial condition*. *Initial condition* yang digunakan adalah $x = 1$ m. *Sliding surface* yang dihasilkan oleh PID-SMC ditampilkan pada Gambar 4.21.

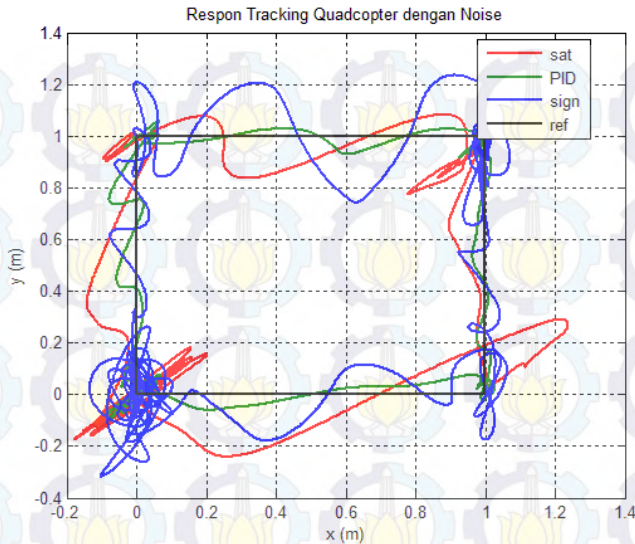


Gambar 4.21 *Sliding Surface* Gerak Sumbu X dengan PID-SMC

Dari Gambar 4.21, dapat disimpulkan bahwa *Sliding Mode Control* berbasis PID mampu memaksa *state* dari sistem untuk mengikuti trajektori yang direncanakan

4.2.5. Simulasi *Sliding Mode Control* dengan *Noise*

Pada pengujian selanjutnya akan dilakukan simulasi *Sliding Mode Control* dengan fungsi *signum* untuk gerak *Waypoint Tracking* dengan *noise*. *Noise* yang akan disimulasikan berupa *noise* pada sinyal kontrol. *Noise* yang akan disimulasikan berupa sinyal *random*. Pada pengujian dengan *noise* berupa sinyal *random*, sinyal *random* yang digunakan adalah bilangan *random* dengan nilai minimal -0,1 dan nilai maksimal 0,1.



Gambar 4.22 Lintasan *Tracking Quadcopter* dengan *Noise*

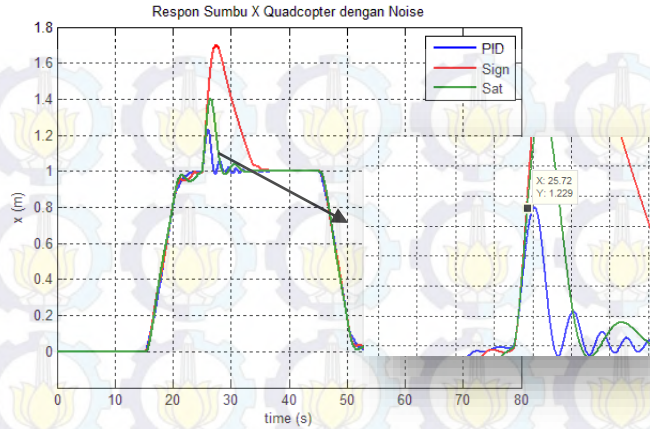
Perbandingan *error* pada tiap-tiap sinyal kontrol natural ditampilkan pada Tabel 4.7.

Tabel 4.7 *Error Tracking Quadcopter* dengan *Noise* pada Sinyal Kontrol

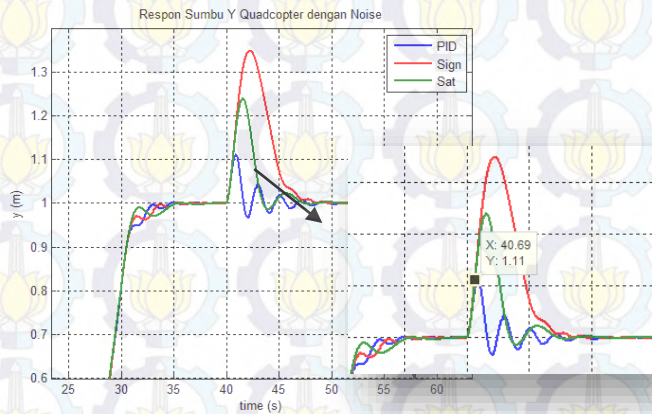
U_N	RMSE X	RMSE Y
sign (s)	9,73 %	30,13%
sat (s)	27,50%	28,46%
$k_p s + k_i \int s dt + k_d \frac{ds}{dt}$	9,44%	8,42%

Dari data yang didapatkan, dapat dilihat bahwa sistem dengan sinyal kontrol natural berupa struktur PID menghasilkan *error* yang paling kecil, yaitu sebesar 9,44% pada sumbu x dan 8,42% pada sumbu y.

Apabila digunakan *noise* berupa sinyal Gaussian, maka respon sistem ditunjukkan pada Gambar 4.23 dan Gambar 4.24.



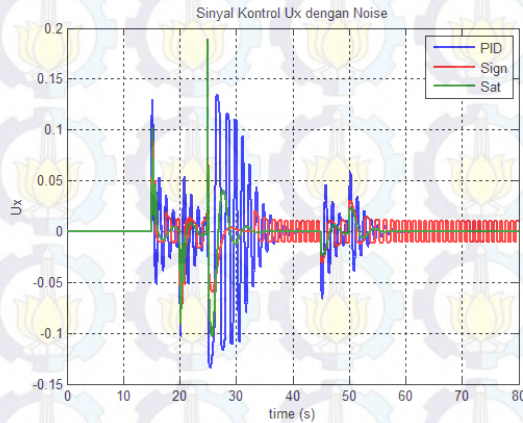
Gambar 4.23 Respon Gerak Sumbu X *Quadcopter* dengan Noise



Gambar 4.24 Respon Gerak Sumbu Y *Quadcopter* dengan Noise

Dimana *noise* Gaussian yang digunakan mempunyai magnitude 0,2 dengan fungsi *noise* eksponensial.

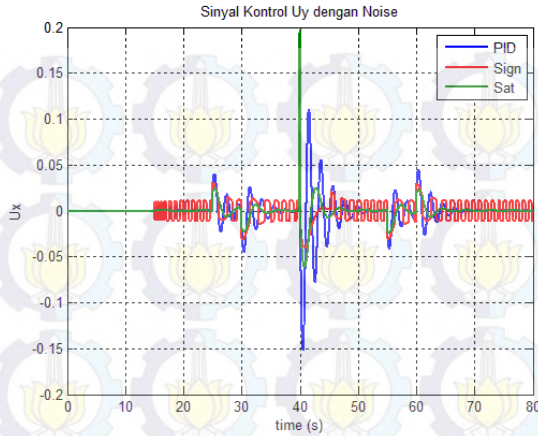
Sinyal kontrol yang dihasilkan pada gerak di sumbu x ditunjukkan pada Gambar 4.25.



Gambar 4.25 Sinyal Kontrol U_x dan U_y pada Gerak Sumbu X dengan Noise

Dari Gambar 4.25, dapat dilihat bahwa *Sliding Mode Control* dengan struktur PID menghasilkan aksi kontrol yang lebih besar, sehingga posisi *quadcopter* dapat kembali ke asalnya dengan lebih cepat. Sedangkan, *Sliding Mode Control* dengan fungsi *signum* menghasilkan sinyal kontrol yang lebih kecil sehingga posisi *quadcopter* kembali ke asalnya lebih lambat.

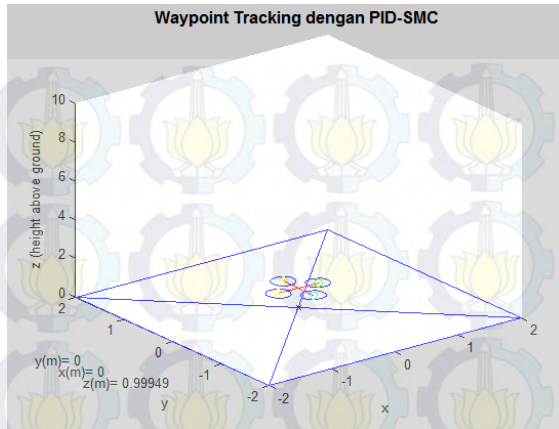
Sinyal kontrol yang dihasilkan pada gerak di sumbu y ditunjukkan pada Gambar 4.26.



Gambar 4.26 Sinyal Kontrol U_x dan U_y pada Gerak Sumbu Y dengan *Noise*

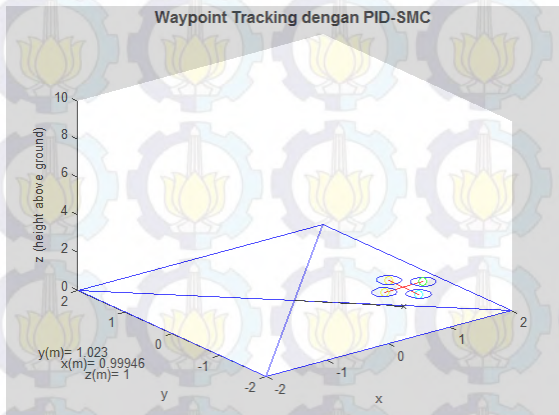
4.3. Simulasi 3D untuk Gerak *Waypoint Tracking* pada *Quadcopter*

Pada pengujian simulasi, terdapat simulasi 3D yang dapat menunjukkan penerbangan *quadcopter* secara visual. Simulasi ini dapat digunakan untuk mengetahui gerakan yang akan terjadi pada *quadcopter* jika diimplementasikan pada plant sesungguhnya. Gambar 4.27, 4.28, dan 4.29 menunjukkan hasil simulasi 3D pengaturan gerak *waypoint tracking* pada *quadcopter*. Input berupa lintasan persegi dengan rentang nilai x, y , dan z masing-masing antara 0-1 m.



Gambar 4.27. Simulasi 3D *Quadcopter* pada Posisi (0;0;0,99949) m

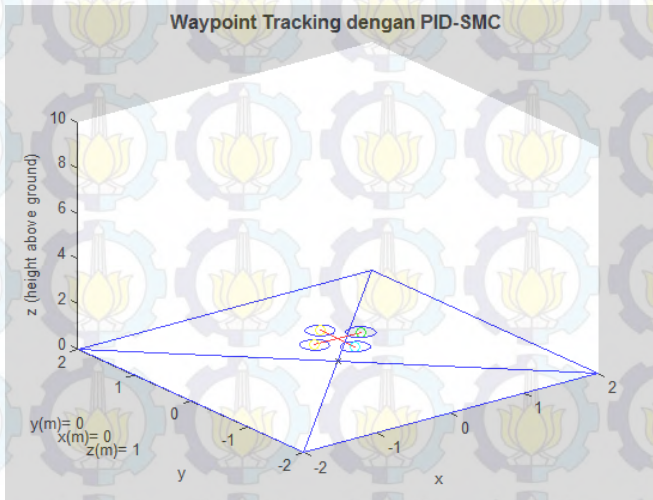
Gambar 4.28 menunjukkan posisi *quadcopter* dititik (0;0;0,9949). Pada gambar ini, *quadcopter* belum melakukan gerak *tracking*, namun telah melakukan *take-off*. *Quadcopter* akan terbang secara vertikal menuju ketinggian 1 m.



Gambar 4.28 Simulasi 3D *Quadcopter* untuk pada posisi (1.023,0.99946,1) m

Gambar 4.28 menunjukkan posisi *quadcopter* telah berada diketinggian 1 m dengan koordinat (1,023;0,99946;1). Sinyal referensi

pada posisi x dan y yang diberikan bukan merupakan sinyal *step*, namun merupakan sinyal *ramp*, untuk menghindari kecepatan yang sangat tinggi dari *quadcopter* saat bergerak.



Gambar 4.29 Simulasi 3D *Quadcopter* pada posisi $(0;0;1)$ m

Gambar 4.29 menunjukkan *quadcopter* telah selesai melakukan proses *tracking*, sehingga berada pada koordinat $(0;0;1)$.

BAB V PENUTUP

Dari penelitian yang dilakukan, didapatkan kesimpulan dan saran sebagai berikut :

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis, dapat diperoleh kesimpulan sebagai berikut:

- a) Metode *PID-Based Sliding Mode Control* yang diaplikasikan pada *quadcopter* dapat digunakan untuk *tracking* dengan lintasan persegi.
- b) Metode *PID-Based Sliding Mode Control* yang diaplikasikan pada *quadcopter* dapat digunakan *tracking* posisi *quadcopter* dengan lintasan yang diinginkan. Untuk *tracking* sumbu x dan y , didapatkan *error* RMS masing-masing sebesar 0,2% dan 0,47%.
- c) Metode *Sliding Mode Control* berbasis PID dapat mengatasi gangguan berbentuk sinyal *Gaussian* yang diberikan pada sinyal kontrol pada gerak *waypoint tracking quadcopter*.
- d) Kontroler yang telah dirancang belum dapat diimplementasikan karena Ardupilot 2.6 tidak dapat diubah *script*-nya dan belum dapat membentuk komunikasi antara MATLAB dengan kontroler.

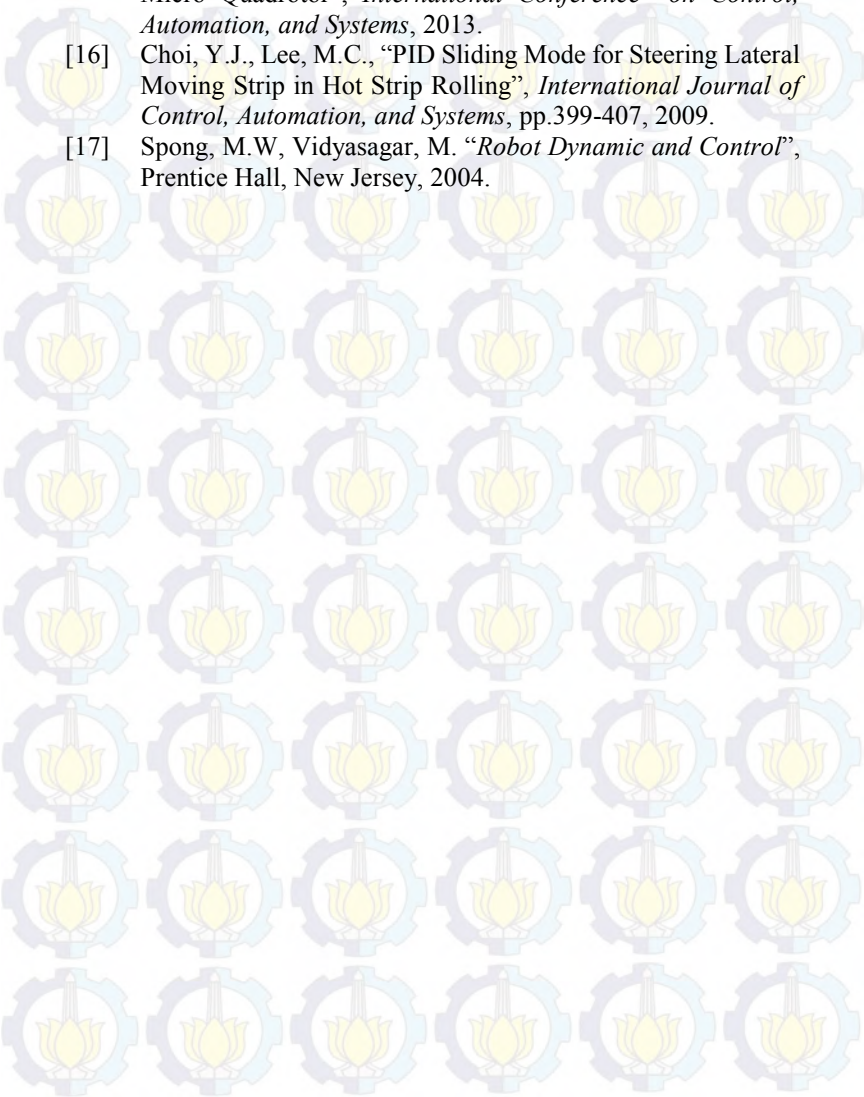
5.2. Saran

Dari hasil penelitian yang dilakukan, untuk pengembangan berikutnya, disarankan beberapa hal berikut ini:

- a) Pemodelan dan pemahaman tentang *quadcopter* akan membantu perancangan sistem yang lebih baik dan mendapatkan model matematika yang lebih akurat.
- b) Perancangan mekanik dan pengetahuan program yang lebih baik akan memudahkan implementasi algoritma kontrol yang akan digunakan dan dianalisis.
- c) Nilai k_p , k_i , dan k_d sebaiknya dilakukan *tuning* secara adaptif supaya kestabilan dari sistem selalu terjaga.

DAFTAR PUSTAKA

- [1] Bouabdallah, S., Murrieri, P. and Siegwart, R., "Towards Autonomous Indoor Micro VTOL", *Autonomous Robots*, vol. 18, 2005.
- [2] Domingues, Jorge Miguel Brito, "Quadrotor Prototype", Universidade Tecnica de Lisboa, *Thesis*. 2009.
- [3] Tommaso Bresciani, "Modelling, Identification and Control of a Quadrotor Helicopter". Department of Automatic Control Lund University, *Thesis*, 2008.
- [4] ..., "Ardupilot Datasheet", <URL : <http://www.ardupilot.co.uk>>, Januari, 2016.
- [5] ..., "FrSky Remote Control Datasheet", <URL : <http://www.frsky-rc.com>>, Januari, 2016.
- [6] ..., "Bulletproof ESC 30A Datasheet", <URL : <http://team-blacksheep.com>>, Januari, 2016.
- [7] ..., "SunnySky 980 kV Datasheet", <URL : <http://www.buddyrc.com/sunnysky-x2212-13-980kv-ii.html>>, Januari, 2016.
- [8] Slotine, J.E. and Li, W., "Applied Nonlinear Control", Prentice Hall, New Jersey, 1991.
- [9] Young, K., V. Utkin and Ü. Özgüner, "A Control Engineers Guide to Sliding Mode Control", *IEEE Transactions On Control Systems Technology*, pp. 328-342, 1999.
- [10] Ogata, K., "Modern Control Engineering", Prentice Hall, New Jersey, 1997.
- [11] DeCarlo, R.A., S.H. Zak and G.P. Matthews, "Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial." *Proceedings of the IEEE*, 1988.
- [12] Khalil, H., "Nonlinear Systems", Prentice Hall, 1996.
- [13] Isidori, A. "Nonlinear Control Systems: An Introduction". Berlin: SpringerVerlag, 1989.
- [14] M. Li, F. Wang and F. Gao, "PID - Based Sliding Mode Controller for Nonlinear Processes ", *Industrial & Engineering Chemistry Research*, Vol. 40, No. 12, pp. 2660 - 2667, 2001.

- 
- [15] Long, T. L., Won Sangchul, “PID Based Sliding Mode for The Micro Quadrotor”, *International Conference on Control, Automation, and Systems*, 2013.
- [16] Choi, Y.J., Lee, M.C., “PID Sliding Mode for Steering Lateral Moving Strip in Hot Strip Rolling”, *International Journal of Control, Automation, and Systems*, pp.399-407, 2009.
- [17] Spong, M.W, Vidyasagar, M. “*Robot Dynamic and Control*”, Prentice Hall, New Jersey, 2004.

LAMPIRAN A

Pengukuran Kecepatan Motor

Pulsa	Motor 1	Motor 2	Motor 3	Motor 4
75	114,505	111,888	109,376	113,668
80	152,29	152,604	148,836	154,069
85	187,563	192,900	187,772	193,633
90	219,276	225,242	221,684	229,429
95	252,77	258,108	256,747	263,76
100	281,03	290,973	287,31	295,578
105	313,267	325,722	322,373	330,223
110	342,26	355,552	351,156	360,262
115	369,892	384,964	380,986	390,93
120	399,198	416,468	412,386	422,958
125	427,04	447,031	440,646	452,16
130	453,625	471,732	466,29	475,396
135	479,164	500,411	496,957	508,575
140	502,295	503,97	519,042	533,172
145	525,531	523,752	543,848	558,292
150	546,569	548,872	566,037	582,784
155	570,538	572,212	574,410	606,334
160	590,529	595,344	592,832	630,721
165	611,567	615,126	609,997	654,271
170	633,024	629,465	627,790	656,050
175	651,236	653,434	650,817	661,702
180	673,111	670,076	672,378	673,844
185	676,356	672,169	675,937	676,565

Pulsa	Motor 1	Motor 2	Motor 3	Motor 4
190	672,902	672,483	673,00	675,1
195	673,006	674,053	673,948	675,832
200	673,006	674,262	672,797	675,623

Kecepatan dalam rad/s

LAMPIRAN B

Penghitungan Konstanta *Drag* dan *Thrust*

Konstanta *drag* dan *thrust* dihitung dengan menggunakan data pengukuran gaya angkat dari quadcopter. Konstanta *thrust* dihitung dengan menggunakan rumus sebagai berikut :

$$b = \frac{m \cdot g}{\sum_{i=1}^4 \Omega_i^2}$$

Dengan nilai F adalah gaya angkat, g adalah percepatan gravitasi, dan Ω adalah kecepatan motor.

Sedangkan untuk menghitung konstanta *drag*, quadcopter diterbangkan, dan digunakan persamaan berikut :

$$\Sigma F = Thrust - mg - Drag$$

$$CD = \frac{D}{\Omega^2}$$

Data pengukuran gaya angkat terhadap pulsa adalah sebagai berikut :

Pulsa	Gaya Angkat 1 (kg)	Gaya Angkat 2 (kg)	Gaya Angkat 3 (kg)	Gaya Angkat 4 (kg)
75	0,02	0,011	0,017	0,014
80	0,036	0,031	0,033	0,039
85	0,058	0,055	0,055	0,061
90	0,078	0,08	0,077	0,084
95	0,104	0,107	0,102	0,115
100	0,131	0,138	0,134	0,145
105	0,165	0,177	0,17	0,184
110	0,198	0,207	0,209	0,221
115	0,227	0,252	0,243	0,264
120	0,27	0,297	0,285	0,306

Pulsa	Gaya Angkat 1 (kg)	Gaya Angkat 2 (kg)	Gaya Angkat 3 (kg)	Gaya Angkat 4 (kg)
125	0,31	0,34	0,328	0,357
130	0,352	0,39	0,374	0,402
135	0,396	0,432	0,426	0,452
140	0,438	0,446	0,478	0,498
145	0,488	0,475	0,522	0,553
150	0,533	0,54	0,578	0,611
155	0,582	0,605	0,613	0,664
160	0,63	0,626	0,619	0,718
165	0,672	0,654	0,646	0,771
170	0,723	0,715	0,704	0,783
175	0,772	0,784	0,762	0,791
180	0,834	0,832	0,826	0,825
185	0,842	0,846	0,84	0,847
190	0,833	0,833	0,836	0,841
195	0,832	0,834	0,834	0,845
200	0,834	0,836	0,837	0,845

LAMPIRAN C

Identifikasi fisik konstanta

$$I_{z,frame1,4} = \frac{m_l}{12}(p^2 + l^2)$$

$$I_{z,frame2,3} = \frac{m_l}{12}(p^2 + t^2)$$

$$I_{z,m1,2,3,4} = \frac{m_m}{2} r^2 + (p_l)^2, M$$

$$I_{z,batt} = \frac{m_b}{12}(p^2 + l^2)$$

$$I_{z,1} = \frac{m_{mika}}{12}(p^2 + l^2)$$

$$I_{z,2} = \frac{m_{mika}}{12}(2, l^2)$$

$$I_{z,1,2,3,4} = \frac{m_p}{2} r^2 + (p_l)^2, M_p$$

$$I_{x,frame1,2,3} = \frac{m_l}{12}(l^2 + t^2)$$

$$I_{x,frame4,5,6} = \frac{m_l}{12}(p^2 + l^2)$$

$$I_{z,m2,4} = m_m \left(\frac{r^2}{4} + \frac{h}{12} \right) + [(0,5h)^2 + (p_1 - p_2)^2] m$$

$$I_{z,m1,3} = m_m \left(\frac{r^2}{4} + \frac{h}{12} \right) + (0,5h)^2 m$$

$$I_{x,batt} = \frac{m_b}{12}(l^2 + t^2) + (0,5t)^2$$

$$I_{y,batt} = \frac{m_b}{12}(p^2 + t^2) + (0,5t)^2$$

$$I_{x,mika1} = \frac{m_{mika}}{12}(p^2 + t^2) + (t^2) m_{mika}$$

$$I_{x,mika2,3} = \frac{m_{mika}}{12}(l^2 + t^2) + (t^2) m_{mika}$$

$$I_{y,mika1} = \frac{m_{mika}}{12}(l^2 + t^2) + (t^2) m_{mika}$$

$$I_{x,mika2,3} = \frac{m_{mika}}{12}(l^2 + t^2) + (t^2 + l^2) m_{mika}$$

$$I_{x,1,2} = m_p \left(\frac{p_2^2}{6} + \frac{t^2}{12} + p_1^2 + (0,5t^2 + h)^2 \right)$$

$$I_{x,3,4} = m_p \left(\frac{p_2^2}{6} + \frac{t^2}{12} \right)$$

Perhitungan I_{xx} , I_{yy} , I_{zz} dan J_{TP}

$$I_{xx} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12}$$

$$I_{xx} = 1,68 \times 10^{-3}$$

$$I_{yy} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12}$$

$$I_{yy} = 1,68 \times 10^{-3}$$

$$I_{zz} = \frac{MR^2}{2} + 4mr^2$$

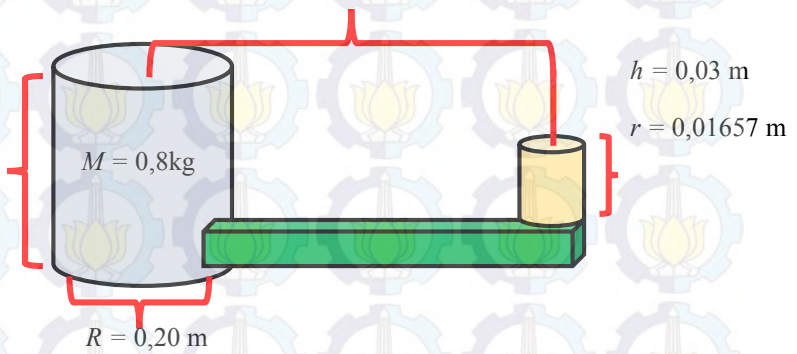
$$I_{zz} = 1,25 \times 10^{-3}$$

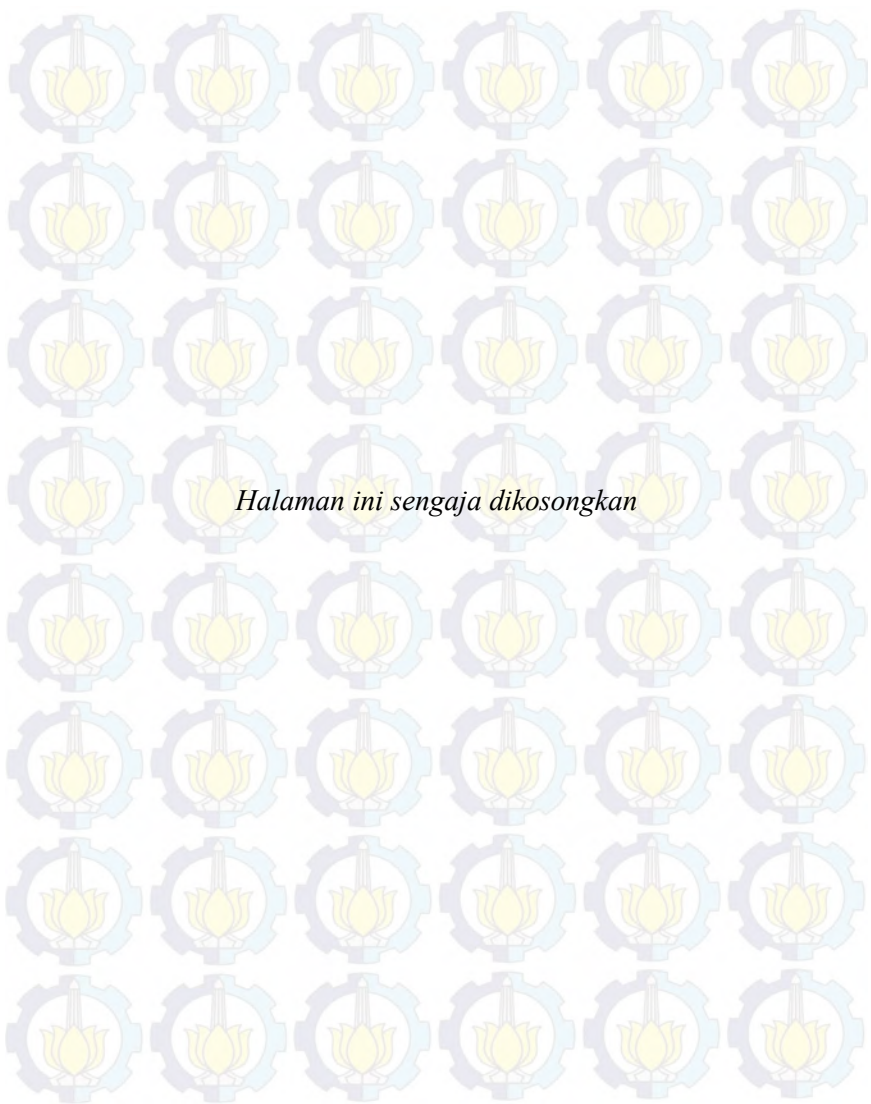
$$J_{TP} = \frac{4 \times (mr^2)}{2}$$

$$J_{TP} = 0,21175 \times 10^{-4}$$

$$l = 0,206 \text{ m}$$

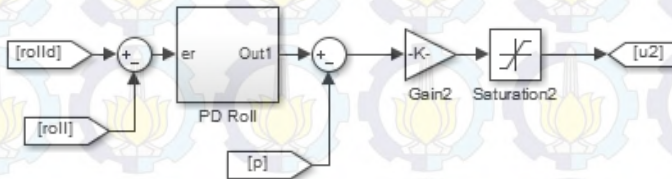
$$H = 0,061 \text{ m}$$



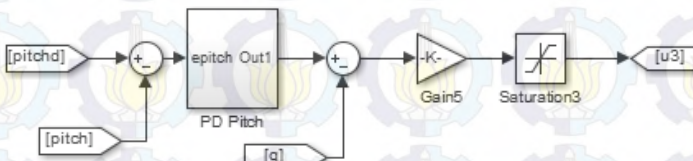


LAMPIRAN D

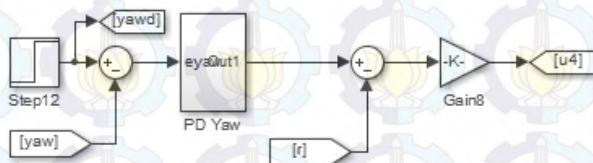
Blok Diagram Simulink
Simulink Kontrol Rotasi *Roll*



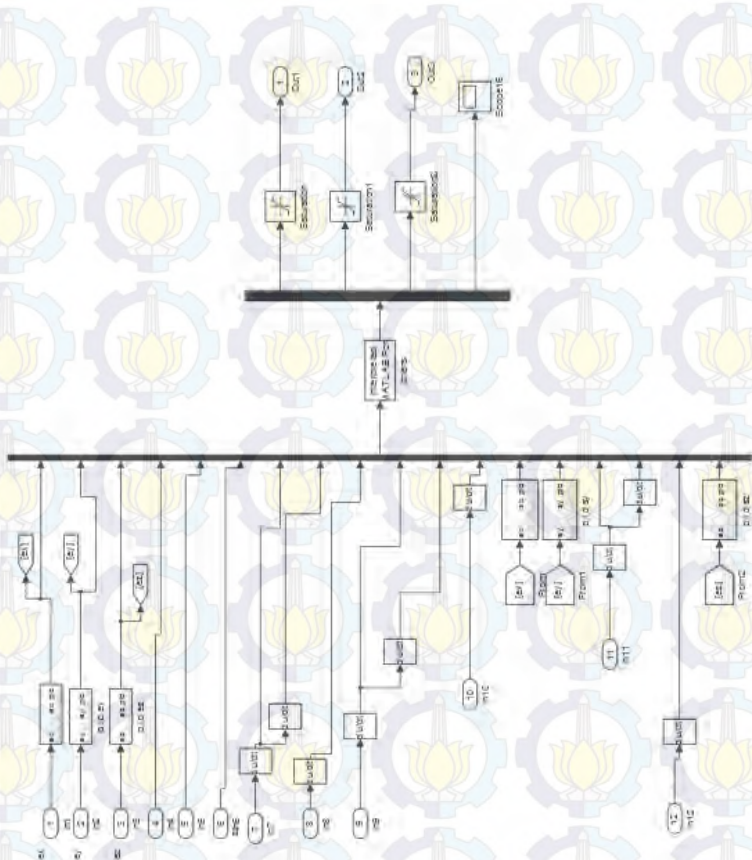
Simulink Kontrol Rotasi *Pitch*



Simulink Kontrol Rotasi *Yaw*



Simulink Kontrol Translasi



```
function out=control_trans(in)
ex = in(1);
iex= in(2);
dex= in(3);
ey = in(4);
iey= in(5);
dey= in(6);
```

```

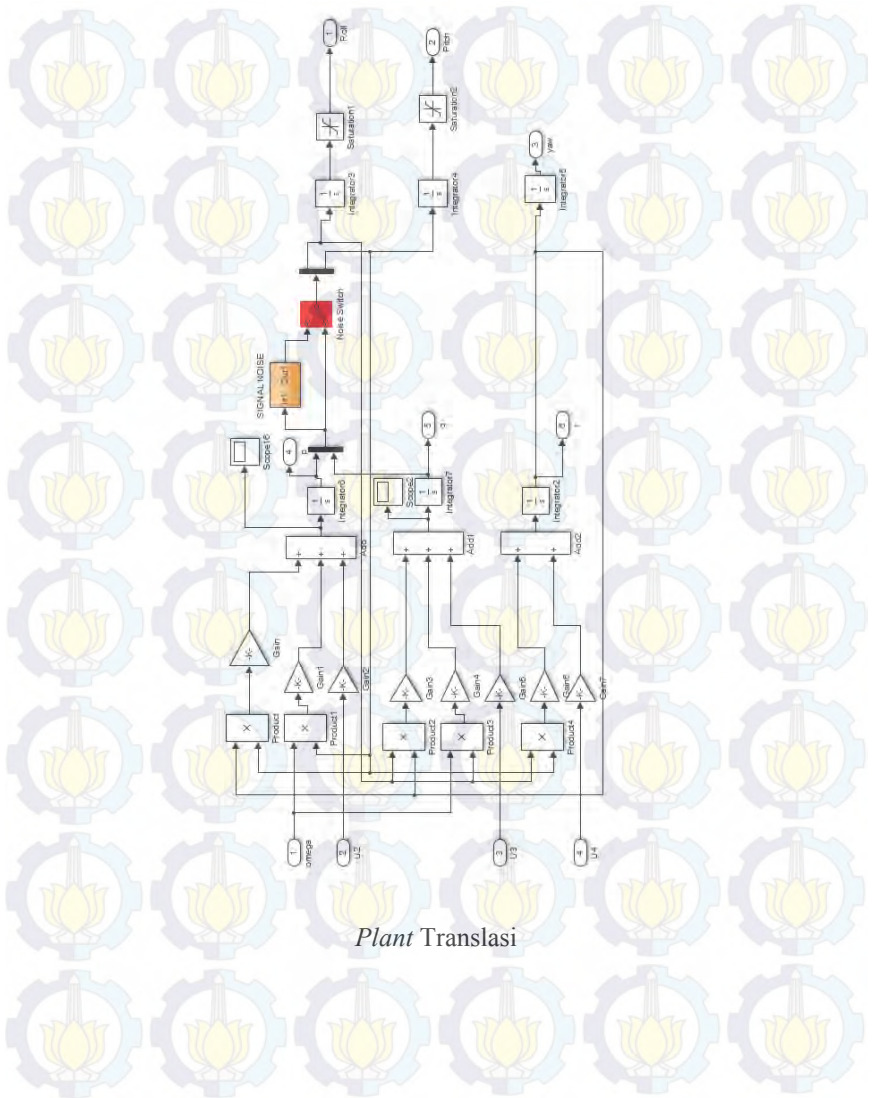
ez = in(7);
iez= in(8);
dez= in(9);
pitch= in(10);
roll=in(11);
yaw=in(12);
dzd=in(13);
ddzd=in(14);
dz=in(15);
dxd=in(16);
ddxd=in(17);
dx=in(18);
sxpид=in(19);
sypид=in(20);
dyd=in(21);
ddyд=in(22);
dy=in(23);
szpid=in(24);
g=9.81;

uz=(g+ ddzd +dez +
szpid)/(cos(roll)*cos(pitch));
ux=1/uz*(ddxd + dex + sxpid);
uy=1/uz*(ddyд + dey + sypид);

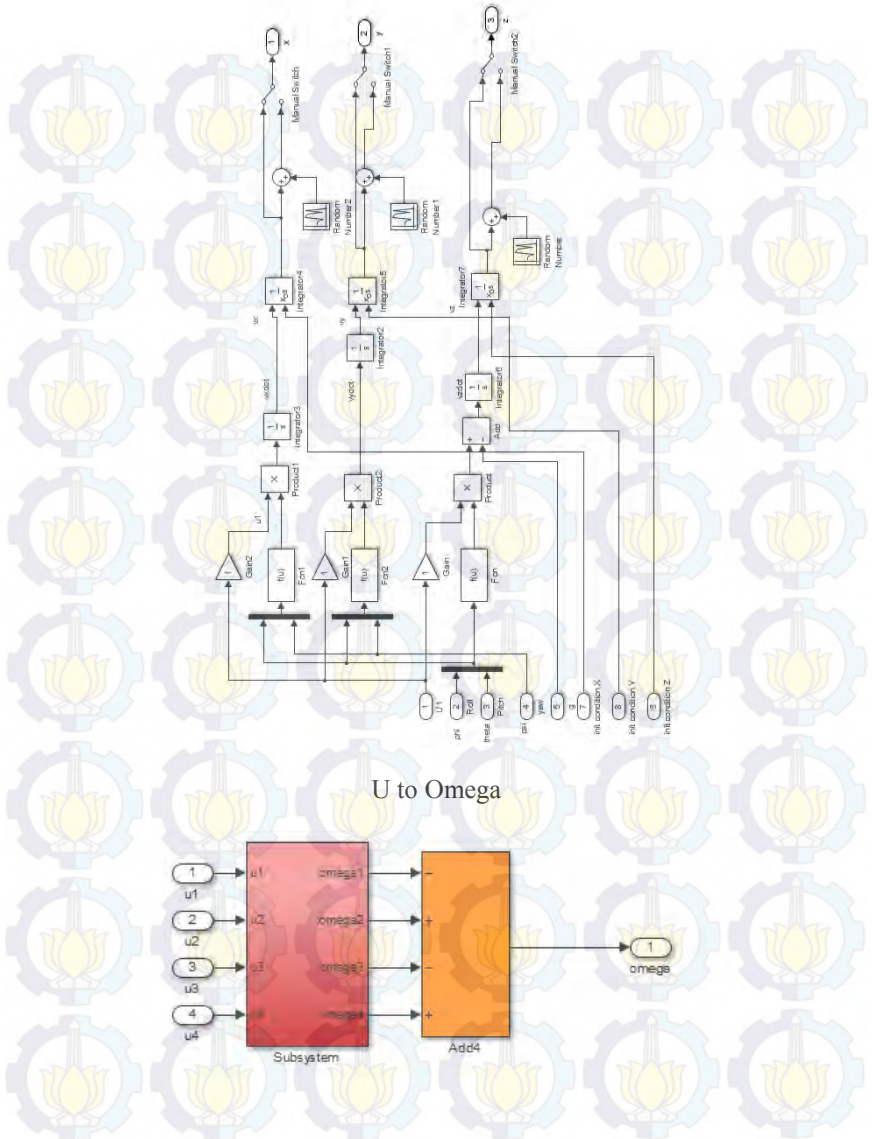
%output
out (1) = ux;%pitch
out (2) = uy;%roll
out (3) = uz;%thrust
out (4) = sign(szpid);
end

```

Plant Rotasi



Plant Translasi



```
function out=u_omega(in)
```

```

u1=in(1);
u2=in(2);
u3=in(3);
u4=in(4);

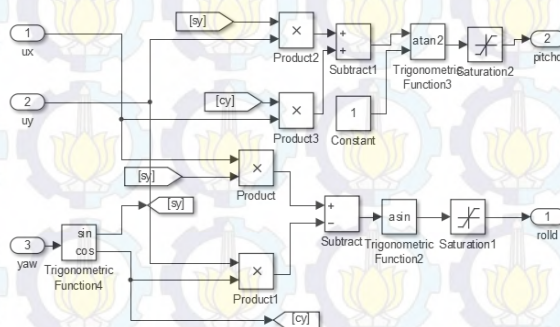
b=1.6798*10^-5;
l=0.206;
d=4.19*10^-6;

omega1=sqrt(abs(u1/(4*b)-u3/(2*b*l)-
u4/(4*d)));
omega2=sqrt(abs(u1/(4*b)-
u2/(2*b*l)+u4/(4*d)));
omega3=sqrt(abs(u1/(4*b)+u3/(2*b*l)-
u4/(4*d)));
omega4=sqrt(abs(u1/(4*b)+u2/(2*b*l)+u4/(4*d)
));

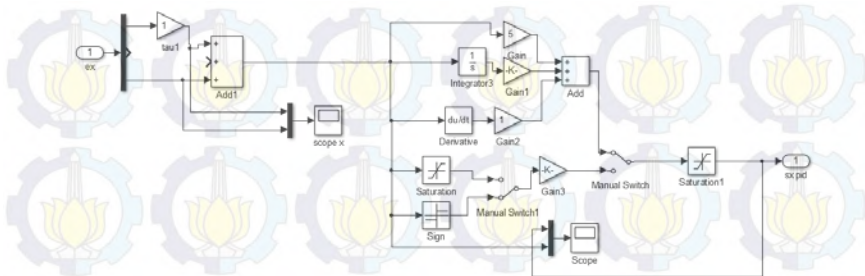
out(1)=omega1;
out(2)=omega2;
out(3)=omega3;
out(4)=omega4;

```

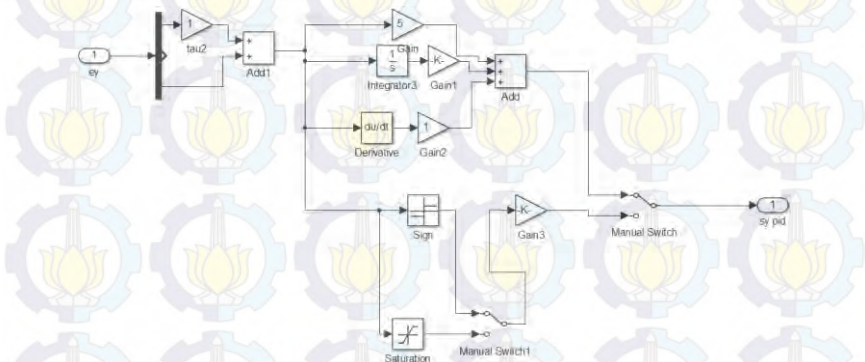
U to sudut



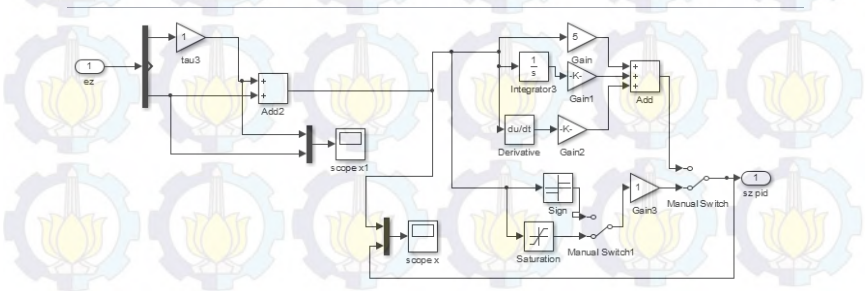
Sinyal Kontrol Natural SMC Sumbu x



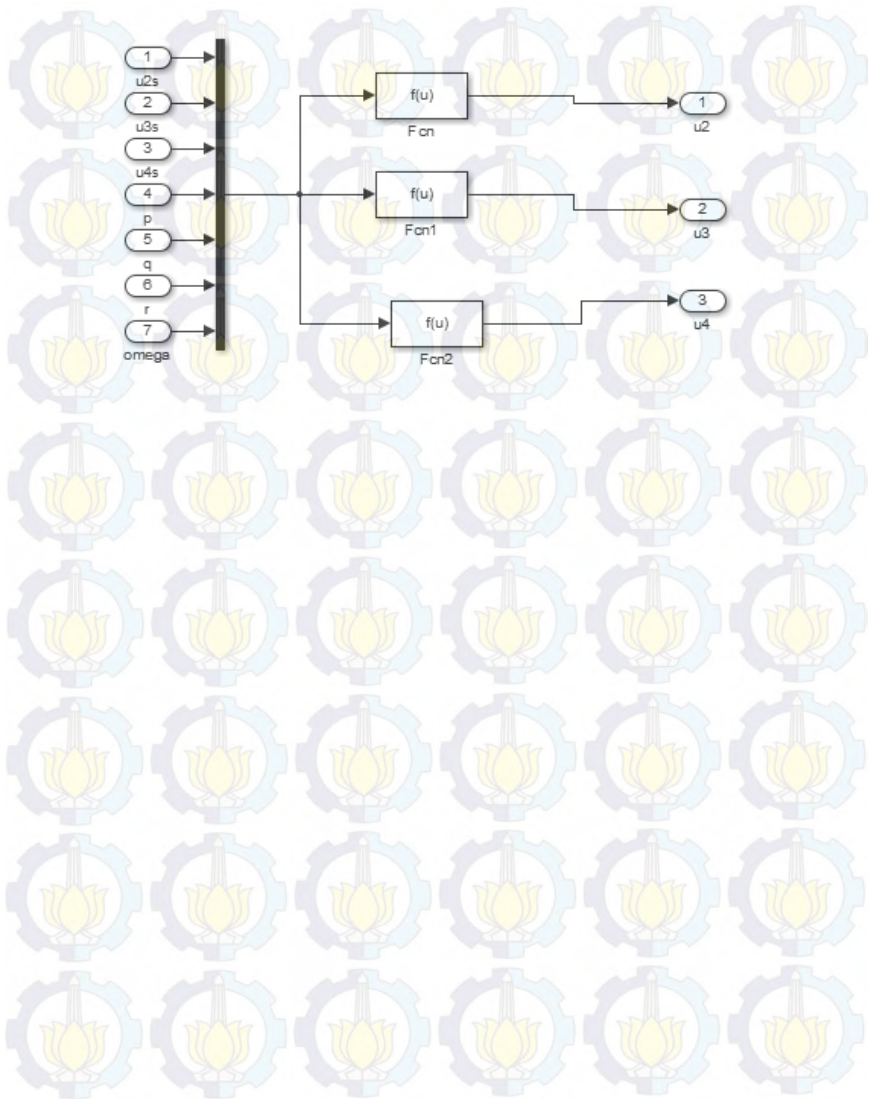
Sinyal Kontrol Natural SMC Sumbu y



Sinyal Kontrol Natural SMC Sumbu z



Linearisasi Sinyal Kontrol



RIWAYAT PENULIS



Penulis bernama lengkap Yurid Eka Nugraha, lahir di Jombang pada tanggal 15 Februari 1995. Penulis memulai jenjang pendidikan dasar di SDN Sukodono I pada tahun 2000, kemudian melanjutkan pendidikannya di SMPN 2 Taman dan SMAN 4 Sidoarjo. Pendidikan tingkat Sarjana ditempuh penulis pada tahun 2012 di Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, Jawa Timur. Pada bulan Januari 2016, penulis mengikuti seminar dan ujian tugas akhir di Bidang Studi Teknik Sistem Pengaturan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana.